

**AD-A234 893**



2

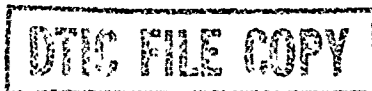
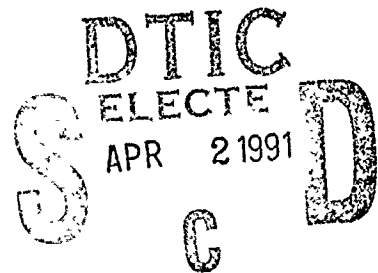
**RADC-TR-90-404, Vol XIV (of 18)**  
**Final Technical Report**  
**December 1990**



# **KNOWLEDGE BASE RETRIEVAL USING PLAUSIBLE INFERENCE**

**Northeast Artificial Intelligence Consortium (NAIC)**

**W. Bruce Croft and Paul R. Cohen**



*APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.*

**This effort was funded partially by the Laboratory Director's fund.**

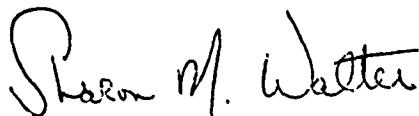
**Rome Air Development Center  
Air Force Systems Command  
Griffiss Air Force Base, NY 13441-5700**

**91 4 11 014**

This report has been reviewed by the RADC Public Affairs Division (PA) and is releasable to the National Technical Information Services (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-90-404, Volume XIV (of 18) has been reviewed and is approved for publication.

APPROVED:



SHARON M. WALTER  
Project Engineer

APPROVED:



RAYMOND P. URTZ, JR.  
Technical Director  
Directorate of Command & Control

FOR THE COMMANDER:



IGOR G. PLONISCH  
Directorate of Plans & Programs

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (COES) Griffiss AFB NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)		2. REPORT DATE December 1990		3. REPORT TYPE AND DATES COVERED Final Sep 84 - Dec 89	
4. TITLE AND SUBTITLE KNOWLEDGE BASE RETRIEVAL USING PLAUSIBLE INFERENCE				5. FUNDING NUMBERS C - F30602-85-C-0008 PE - 62702F PR - 5581 TA - 27 WU - 13 (See reverse)	
6. AUTHOR(S) W. Bruce Croft and Paul R. Cohen				8. PERFORMING ORGANIZATION REPORT NUMBER N/A	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Northeast Artificial Intelligence Consortium (NAIC) Science & Technology Center, Rm 2-296 111 College Place, Syracuse University Syracuse NY 13244-4100				10. SPONSORING/MONITORING AGENCY REPORT NUMBER RADC-TR-90-404, Vol XIV (of 18)	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Rome Air Development Center (COES) Griffiss AFB NY 13441-5700				(See reverse)	
11. SUPPLEMENTARY NOTES RADC Project Engineer: Snaron M. Walter/COES/(315) 330-3577 This effort was funded partially by the Laboratory Director's fund.					
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The Northeast Artificial Intelligence Consortium (NAIC) was created by the Air Force Systems Command, Rome Air Development Center, and the Office of Scientific Research. Its purpose was to conduct pertinent research in artificial intelligence and to perform activities ancillary to this research. This report describes progress during the existence of the NAIC on the technical research tasks undertaken at the member universities. The topics covered in general are: versatile expert system for equipment maintenance, distributed AI for communications system control, automatic photointerpretation, time-oriented problem solving, speech understanding systems, knowledge base maintenance, hardware architectures for very large systems, knowledge-based reasoning and planning, and a knowledge acquisition, assistance, and explanation system.  The specific topic for this volume is plausible inference as an effective computational framework for the retrieval of complex objects.					
14. SUBJECT TERMS Artificial Intelligence, Knowledge Based, Intelligent Retrieval, Plausible Inference				15. NUMBER OF PAGES 68	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL		

Block 5 (Cont'd)      Funding Numbers

PE - 62702F	PE - 61102F	PE - 61102F	PE - 33126F	PE - 61101F
PR - 5581	PR - 2304	PR - 2304	PR - 2155	PR - LDFP
TA - 27	TA - J5	TA - J5	TA - 02	TA - 27
WU - 23	WU - 01	WU - 15	WU - 10	WU - 01

Block 11 (Cont'd)

This effort was performed as a subcontract by the University of Massachusetts at Amherst to Syracuse University, Office of Sponsored Programs.

Volume 14

1989 ANNUAL REPORT  
To Rome Air Development Center

*Knowledge Base Retrieval Using Plausible Inference*

W. Bruce Croft  
Paul. R. Cohen

Department of Computer and Information Science  
University of Massachusetts  
Amherst, Massachusetts

Accession For	
EX-101	<input checked="" type="checkbox"/>
EX-102	<input type="checkbox"/>
EX-103	<input type="checkbox"/>
EX-104	<input type="checkbox"/>
By	
Dispers.	
Availability Code	
Dist	Special
A-1	

## Contents

14.1	Executive Summary . . . . .	3
14.2	Ancillary Activities . . . . .	5
14.2.1	Degrees Conferred . . . . .	5
14.2.2	New AI Faculty . . . . .	5
14.2.3	New AI Courses . . . . .	5
14.2.4	Journals, Book Chapters, and Conference Papers . . . . .	5
14.2.5	Progress or Products Resulting from NAIC Research . . . . .	6
14.2.6	Improvements to Research Environment . . . . .	6
14.3	Overview of Research . . . . .	7
14.3.1	An Experimental Study of Retrieval Using Plausible Inference	10
14.3.2	A Retrieval Model for Complex Documents . . . . .	19
14.3.3	Natural Language Processing and Information Retrieval . . .	25
14.3.4	Structures for Plausible Inference in Semantic Networks . . .	40
14.3.5	Future Directions . . . . .	49

## 14.1 Executive Summary

One of the primary functions in many knowledge-based applications is the retrieval of objects that satisfy criteria specified in a user's query. Examples of objects that may be retrieved in this way include natural language documents or parts of documents, and fragments of encyclopedic knowledge bases. In these cases, the process of determining if a query criterion is satisfied will involve inference. One approach to this problem would be to represent objects and knowledge about objects in a deductive database system. In such a system, a query can be expressed in the form  $q = \{X|W(X)\}$  where  $X$  is a vector of domain variables and  $W(X)$  is a formula in which  $X$  are the only free variables. Retrieval involves finding all instances of  $X$  for which  $W(X)$  can be proved. In other words, for a query  $q$ , retrieve  $X$  where  $KB \models W(X)$ . The knowledge base,  $KB$ , includes descriptions of objects (extensional data), rules (intensional data), and basic axioms. The main issue in implementing deductive database systems is designing efficient inference methods.

The critical issue we have studied is the *effectiveness* of retrieval. By this, we mean how well the system does at locating objects that are judged relevant by the user. Designing effective retrieval strategies is difficult because in real environments the query specification, the object descriptions, and the rules in the knowledge base are incomplete and uncertain. In these situations involving uncertain information, *deductive inference does not provide effective retrieval*. Instead, retrieval must be implemented as a process of plausible inference or evidential reasoning.

The aim of our research is to demonstrate that plausible inference is an effective computational framework for retrieval of complex objects. To do this, we must address a few key issues. In particular, we must describe an appropriate formal model of plausible inference, show how this model is implemented in a real system, and evaluate its performance. In the relatively short duration of this project (14 months), we have begun to make significant progress in all these areas.

In particular, we have pursued both numerical and symbolic approaches to representing and reasoning with uncertainty. Recently, we have been investigating a Bayesian network approach due to the fact that simpler probabilistic models have produced good results in previous research.

We have also been constructing an experimental setting for our research with test collections that consist of a large number of text objects, a linguistic knowledge base, and a domain knowledge base that describes objects and relationships in the domain of the texts. These test collections also must have associated queries and relevance judgements. Some retrieval experiments using heuristic versions of the plausible inference model have been carried out with encouraging results.

We have also concentrated on developing techniques for automatically pro-

ducing complex representations of the meaning of text objects. These natural language processing techniques provide rich sources of evidence for our plausible inference models, as well as providing the basis for important applications of text-based intelligent systems.

Finally, we have conducted a number of experiments on the derivation of plausible inference rules from knowledge bases.



## 14.2 Ancillary Activities

### 14.2.1 Degrees Conferred

- Ph.D.s: 1
- MSs: 2

### 14.2.2 New AI Faculty

- Professor Rod Grupen

### 14.2.3 New AI Courses

A new course on "Text-Based Intelligent Systems" is being developed and will be offered in the Spring Semester of 1990.

### 14.2.4 Journals, Book Chapters, and Conference Papers

Cohen, P.; Loiselle, C., 1988. "Beyond ISA: Structures for Plausible Inference in Semantic Networks", *Proceedings of AAAI-88*, 415-420.

Croft, W.B.; Lucia, T.J.; Cringean, J.; Willett, P., 1989. "Retrieving Documents by Plausible Inference: An Experimental Study". *Information Processing and Management*, (to appear).

Lewis, D.; Croft, W.B.; Bhandaru, N., 1989. "Language Oriented Information Retrieval", *International Journal of Intelligent Systems*, (to appear).

Gay, L.; Croft, W.B., 1989. "Interpreting Nominal Compounds for Information Retrieval", *Information Processing and Management*, (to appear).

Krovetz, R.; Croft, W.B., 1989. "Word Sense Disambiguation Using a Machine-Readable Dictionary", *Proceedings of the 12th International Conference on Research and Development in Information Retrieval*, 127-136.

Krovetz, R., 1989. "Lexical Acquisition and Information Retrieval", *Proceedings of the Workshop on Computational Lexicography*, (to appear).

Croft, W.B.; Turtle, H., 1989. "A Retrieval Model Incorporating Hypertext Links", *Proceedings of Hypertext 89*, (to appear).

#### **14.2.5 Progress or Products Resulting from NAIC Research**

During the period of this contract, we have made contact with, and received some support from, Digital Equipment Corporation, Thinking Machines Corporation, General Electric, Bell Communications Research, OCLC, Ricoh, and Olivetti.

In July 1989, Croft organized the 12th International Conference on Research and Development in Information Retrieval in Cambridge, MA., which featured a number of sessions on knowledge-based retrieval. It was a very successful conference and had a large number of attendees from industry.

Croft, together with Paul Jacobs from GE and David Waltz from Thinking Machines, is organizing a symposium on "Text-Based Intelligent Systems", at the AAAI Spring Symposium Series in March, 1990.

Croft was one of 3 people invited to a workshop on Information Retrieval at Thinking Machines Corporation, Cambridge, MA.

Croft gave a Distinguished Lecture Series talk to OCLC in Columbus, Ohio.

Lewis (a graduate student supported by RADG) gave an invited talk at a Workshop on the Evaluation of Natural Language Processing Systems.

#### **14.2.6 Improvements to Research Environment**

The Information Retrieval Laboratory has acquired 3 MicroExplorers.

### 14.3 Overview of Research

One of the primary functions in many knowledge-based applications is the retrieval of objects that satisfy criteria specified in a user's query. Examples of objects that may be retrieved in this way include natural language documents or parts of documents [1,4], multimedia objects containing graphics, image and voice as well as text [17], and fragments of encyclopedic knowledge bases for AI applications [8]. In many cases, these applications cannot be handled using conventional database technology because of the complexity of the objects being stored and because the process of determining if a query criterion is satisfied may involve inference. One approach to this problem would be to represent objects and knowledge about objects in a deductive database system [6]. In such a system, a query can be expressed in the form  $q = \{X|W(X)\}$  where  $X$  is a vector of domain variables and  $W(X)$  is a formula in which  $X$  are the only free variables. Retrieval involves finding all instances of  $X$  for which  $W(X)$  can be proved. In other words, for a query  $q$ , retrieve  $X$  where  $KB \models W(X)$ . The knowledge base,  $KB$ , includes descriptions of objects (extensional data), rules (intensional data), and basic axioms. The main issue in implementing deductive database systems is designing efficient inference methods.

The critical issue for us, however, is the *effectiveness* of retrieval. By this, we mean how well the system does at locating objects that are judged relevant by the user. This has been a central focus of the research in information retrieval (IR) and a number of evaluation measures and methodologies have been developed [12]. Less than perfect retrieval is the result of people viewing objects not retrieved by the system as being relevant and viewing some objects that are retrieved (i.e. satisfy the query criteria) as not relevant. As we do retrieval experiments, we quickly realize that the usual situation is that the query specification, the object descriptions, and the rules in the knowledge base are incomplete and often errorful. In these situations involving uncertain information, *deductive inference does not provide effective retrieval*. Instead, retrieval must be implemented as a process of plausible inference or evidential reasoning. The classic example of this problem in IR is the common use of Boolean query formulations and string matching techniques in many commercial systems. It has been shown in a number of experiments that techniques based on probabilistic models are much more effective. There has also been significant evidence for the evidential nature of retrieval in that searches based on different aspects of a text object's representation (e.g. full text, citations, keywords) have been shown to retrieve different sets of relevant objects (for example, [7]).

Our aim is to demonstrate that plausible inference is an effective computational framework for retrieval of complex objects. To do this, we must address

a few key issues. In particular, we must describe an appropriate formal model of plausible inference, show how this model is implemented in a real system, and evaluate its performance. Given a query in the form (as before) of  $\{X|W(X)\}$ , we want to retrieve those  $X$  for which  $W(X)$  is plausibly true. Presumably, we also want to retrieve the most plausible  $X$  first. To do this, we must describe the inference methods that are used to establish and quantify the relationships between instances of  $X$  in the knowledge base and the query. We must also be able to generate or acquire the knowledge about plausible relationships that are used by this process. For example, if the query states  $P1(X)$  and the knowledge base contains  $P2(x1)$ , we need to know if it is plausible to conclude  $P1(x1)$  and thereby retrieve  $x1$ .

A number of approaches have been taken to formalizing plausible inference. These include modified logics [16,18,10] and probability-based theories such as Dempster-Shafer [15] and Bayesian networks [11]. These approaches emphasize numerical measures of uncertainty and provide techniques for combining these measures. In contrast, Cohen has investigated symbolic approaches to reasoning with uncertainty [2,3]. In this research, we have pursued both directions, using numerical approaches to represent uncertainty and symbolic approaches to control the inferences that are made. We have in particular begun to study the effectiveness of the Bayesian network approach due to the fact that we have used simpler Bayesian models in our previous research with good results [12].

We have also been constructing an experimental setting for our research with test collections that consist of a large number of text objects, a linguistic knowledge base, and a domain knowledge base that describes objects and relationships in the domain of the texts. These test collections also must have associated queries and relevance judgements. By focusing on text retrieval as the application for the plausible inference techniques, we are tackling a problem that is recognized both as important and difficult. If significant performance improvements can be obtained using plausible inference in this setting, it will be an impressive demonstration of the significance of the approach. Using text highlights the uncertainty of the object and query representations, as well as providing sources of evidence for retrieval that can be generated automatically by natural language processing. These pieces of evidence will be combined with evidence provided by plausible inference using the domain knowledge base. The basis for this system is the GRANT knowledge base used by Cohen for experiments on controlled spreading activation as an implementation of reasoning with uncertainty [3].

In the few months since we have begun this project, we have made significant progress in number of areas. This progress has been reported in papers listed in section 14.2.4. The results have included experimental validation of the plausible in-

ference approach to retrieval, algorithms for matching complex structures generated by NLP, techniques for using natural language processing as a source of evidence for retrieval, and techniques for deriving plausible inference rules from knowledge bases. In the following sections we will discuss some of these topics in more detail.

The next section describes a series of experiments that tested the effectiveness of using multiple sources of evidence in a retrieval context. The experiments were done using fairly simple forms of evidence derived from test collections of text documents. Specifically, the initial evidence used was statistical data about the likelihood of document representations given particular queries. This evidence was then modified using data about the conditional probabilities of document occurrence (i.e. similarities between documents) and, in some cases, citation data. The results obtained indicate that using multiple sources of evidence, even in this simple case, is the most effective retrieval technique. This result is particularly significant in that this is the first series of experiments that has produced results consistently better than the best statistical ranking technique.

In section 14.3.2, we discuss the Bayesian model of retrieval for complex objects and show how it relates to previous probabilistic models.

In 14.3.3, we indicate how natural language processing (NLP) can be used as part of an overall retrieval system based on plausible inference (the ADRENAL system). In this system, NLP techniques are combined with efficient statistical techniques. Preliminary experiments show that evidence from NLP can be combined with other types of evidence to improve performance.

The last section shows how plausible inference rules can be automatically derived from relations in a knowledge base. Experiments with the GRANT knowledge base indicate that simple structural features such as transitivity are excellent predictors of plausibility.

### 14.3.1 An Experimental Study of Retrieval Using Plausible Inference

One of the interesting results of recent research in information retrieval has been the evidence that different search strategies and representations of document content retrieve different relevant documents for the same query [19,7,25]. Strategies that have similar average performance in terms of recall and precision can have very different performance levels with particular queries and the sets of retrieved documents tend to have low overlap. These observations led to the investigation of techniques for choosing an optimal retrieval strategy for a particular query [21,25]. In these studies, it was found that the properties of a query and the associated relevant documents had no apparent correlation with the search strategies that performed best for that query.

Given this evidence, it appeared that the only practical solution was to provide multiple retrieval strategies and representations in the information retrieval system, together with heuristics for choosing between them. For example, in the I<sup>3</sup>R system [4], both probabilistic and cluster search strategies are used when the user expresses interest in high recall. The results of both strategies are displayed in windows for the user to examine. Citations, which can be regarded as an alternate representation, are used during browsing.

In this section, we describe experiments done using plausible inference as an alternative model of retrieval. The first set of experiments were carried out with the I<sup>3</sup>R system and search strategies. We have chosen to implement plausible inference in I<sup>3</sup>R as a form of *constrained spreading activation* [3]. This activation strategy makes use of the network representation of a document collection that is the basis of the browsing facility in I<sup>3</sup>R.

The second set of experiments tested the effectiveness of combining evidence from probabilistic (or best-match) searches with that from nearest-neighbor searches on a variety of test collections. The combination was done using spreading activation as in the previous experiments.

#### 14.3.1.1 The I<sup>3</sup>R System

The I<sup>3</sup>R document retrieval system implements a three-stage retrieval process. These stages are;

1. Construct a representation of the information problem (the request or query).
2. Retrieve documents using the request.
3. Evaluate the retrieved documents and reformulate the request.

At each stage, the system provides a variety of tools for accomplishing the goal of that stage. The control of the use of these tools is facilitated by organizing them as "knowledge sources" in a blackboard-based system. The scheduler is a knowledge source in this organization whose task is to select which tool is to be used at a particular stage of the search session. The most important knowledge sources used in the current version of I<sup>3</sup>R are as follows;

- The *User Model Builder* collects information about the user to determine if a particular *stereotype* applies. Stereotypes determine the style of interaction, goals of the session and other information. The other major function of the user model builder is to acquire knowledge about the user's domain of interest. This domain knowledge is used to refine the request model.
- The *Request Model Builder* constructs a model of the user's information need. The major part of this task is to obtain an initial query from the user and to perform simple statistical indexing on this query to obtain index terms and weights. A number of different forms of query specification are allowed including natural language, words connected with Boolean operators, and words or phrases with associated weights. The request model builder also obtains relevance judgments on retrieved documents and information about the content of individual relevant documents.
- The *Domain Knowledge Expert* uses the domain knowledge from the user model and the knowledge base to infer concepts that are related to those in the initial query. These concepts are presented to the user for evaluation and eventual inclusion into the request model.
- The *Search Controller* selects and executes formal search strategies. The strategies are based on the probabilistic model and clustering [1]. Retrieved documents are placed in the request model.
- The *Browsing Expert* provides the user with an informal method of finding relevant documents by navigating through the knowledge base. The browsing process can start from a given document, author or index term and follows links to other items in the knowledge base. The knowledge acquired during browsing is used to update the request model and may trigger a formal search strategy. Browsing is essentially a user-directed activity but the system provides recommendations of interesting paths [26].

It can be seen that the current version of I<sup>3</sup>R emphasizes the use of statistical indexing and retrieval techniques. Statistical indexing represents the content of

queries and documents with sets of weighted terms, where the terms are important word stems extracted from the document text and the weights reflect the importance of a term in a particular document and in the collection as a whole. Probabilistic retrieval strategies rank documents in order of their probability of relevance to a particular query. This means that scores are calculated for documents using the weights of query terms present in documents. Retrieval based on clustering also ranks retrieved documents, but queries are compared to representatives of groups of similar documents rather than individual documents. The search controller can also invoke a strategy that uses document citations to find relevant documents.

I<sup>3</sup>R is being extended to include knowledge sources that use natural language processing [20] to produce representations of queries and documents based on case frames. These representations can then be used to generate information for statistical searches or matched directly and the result used to retrieve documents.

From the point of view of the research described here, the important features of I<sup>3</sup>R are that it contains tools to construct different representations of request and document content and there are a variety of search strategies that use these representations to infer the relevance of documents. The experiments described here use these features to test the plausible inference retrieval model.

The I<sup>3</sup>R knowledge base contains descriptions of terms and concepts. Terms are extracted from text by statistical analysis and represent single words or stems. Concepts are specified in the current system through a combination of interactive knowledge acquisition and prespecified thesaurus information. The NLP techniques we are developing also use a prespecified vocabulary of concepts. Documents in the knowledge base have title, abstract and journal attributes and are related to terms, concepts and other documents in a variety of ways. Relationships may be statistical (e.g. document-document similarity measures), bibliographic (e.g. author-document), semantic (e.g. synonym) or may simply indicate which terms and concepts represent their content. The inference rules that use these relationships are specified in the knowledge sources.

The statistical relationships between pairs of documents and terms connect the nearest neighbors according to a similarity measure defined on the term-based content representations. These relationships are used as the basis of the cluster searches and can be regarded as a sort of "shortcut" plausible inference.

Concepts in I<sup>3</sup>R are described by *recognition rules* and relationships to other concepts. This representation is related to that used in the Rubric system [27]. The recognition rules describe the different ways in which a concept can be recognized in a text passage. The relationships that are used are SYNONYM, ISA, INSTANCE, PART-OF, CROSS-REFERENCE and their inverses.



The current knowledge base of I<sup>3</sup>R contains the representations of 3200 document abstracts from the Communications of the ACM. This collection has associated with it a set of queries and relevance judgments that are used for retrieval experiments.

#### 14.3.1.2 Spreading Activation

A spreading activation search in a knowledge base represented as a network consists of the following major steps;

1. Certain nodes are identified as starting points for the search. These nodes are usually specified in the query:
2. All nodes connected by a single link to the starting points are activated. The level of activation is determined by a number of factors including the type of link.
3. The process of activating nodes connected to activated nodes is repeated until some termination condition is met. Typically the activation level decreases with the length of the activation path, and the search can be terminated when the activation levels drop below some threshold. As multiple paths may pass through a single node, the activation level of a node is a function of each of these separate activations.
4. Nodes are retrieved in order of their activation levels.

In the GRANT system [3], this simple activation process, which may rapidly activate large portions of the knowledge base, is constrained according to the types of links in the activation path. For example, following an ISA link to the more general concept and then going down an inverse ISA link to another specialized concept generally does not represent a plausible relationship.

In the I<sup>3</sup>R knowledge base, the constraints on spreading activation specify which types of links are reasonable to use and when. It may be possible, using the appropriate constraints, to start at nodes representing query terms and have all those documents that would be retrieved by the probabilistic search activated at the first stage and those that would be retrieved by the cluster search activated at the second stage. We could also use term and concept links to find documents with related contents to those already found. In the experiments described in the next section, however, we have limited the types of links that are used and have the following specific form of constrained spreading activation;

1. The starting points for the spreading activation are the top-ranked documents from the probabilistic search. We do not make any direct use of nodes representing query terms and the corresponding term-document, term-term, term-concept or concept-concept links.
2. The links that are used for spreading activation are document nearest neighbors and document citations. These links are assumed to represent the strongest plausible relationships between documents. When we use these links, we are asserting that if users are interested in a document A, then they will also be interested in the nearest neighbors and citations of A. This approach has some similarity to previous work on using citations as an alternative representation (e.g. [23]).
3. Citation links are only used from the starting documents. In the remaining cycles of activation, nearest neighbor links are used exclusively. This amounts to asserting that citations of anything other than the starting documents will not be interesting to the users.
4. The activation level is a function of the level of activation of the originating document, the strength of the link, and the current activation level of the activated document.
5. Documents that have been used as part of an activation path are not used again if they are reactivated.

In the following section, we describe experiments done by varying parameters of the spreading activation search just described. The goal of these experiments is to determine whether the multiple sources of evidence used by the spreading activation (or plausible inference) search can lead to better performance.

#### 14.3.1.3 The I<sup>3</sup>R Experiments

The following experiments were done using the CACM document collection, including queries and associated relevance judgments. The benchmark for comparison of effectiveness is the results obtained using a probabilistic search with inverse document frequency weights and term significance weights [1]. This is equivalent to a tf.idf search done with the vector space model [14]. The notation tf.idf refers to the weighting used with document and query terms, which is a combination of importance of a term in an individual document (tf) and importance in the collection as a whole (idf). The starting points used in the spreading activation searches were the top twenty ranked documents obtained with the probabilistic search. A subset

of 41 of the total 52 queries in the collection were used. These were the queries that had relevance judgments and had relevant documents in the top twenty ranked documents.

The first experiment was designed to test if there was any correlation between relevance and evidence provided by multiple sources of evidence. Starting with the top twenty ranked documents from the probabilistic search, we formed the following categories of documents;

1. Documents that were in the top ranked list AND were cited by documents in the top ranked list.
2. Documents that were in the top ranked list AND were connected by nearest neighbor links to documents in the top ranked list.
3. Documents that were in the top ranked list AND were connected by nearest neighbor links AND were cited.

For each of these categories, we calculated the percentage of documents that were relevant. Table 1 shows the results.

Category	Percentage Relevant
Top-ranked probabilistic (P)	26%
Cited documents (C)	14%
Nearest neighbors (N)	18%
P and C	52%
P and N	36%
P and C and N	35%

Table 1: Percentages of relevant documents in different categories

The results show that the cited documents and the nearest neighbor documents, on their own, do not contain a large proportion of relevant documents. The high percentages of relevant documents in the categories of documents that come from multiple sources of evidence (P and C, P and N, P and C and N) indicate that these documents are more likely to be relevant than a document in a category from a single source of evidence (P). These results are not conclusive, because the citations of documents were involved in the original relevance judgments and because documents with the most sources of evidence (P and C and N) did not have the highest likelihood of relevance. To see if the multiple sources of evidence as defined

here could be used effectively, the following experiments were done with spreading activation searches.

The spreading activation searches use the top ranked documents from the probabilistic search strategy as starting points. The first set of results were obtained using only a single plausible inference. That is, given the starting documents, we spread activation along a single nearest neighbor or citation link. Documents which are activated have their scores adjusted according to the following algorithm;

1. The new activation level (NACT) of the activated document is determined as a function of the score of the starting document (SACT). In these experiments,  $NACT = \alpha \times SACT$  where  $\alpha$  is a parameter that depends on the type of link ( $\alpha_{cite}, \alpha_{nn}$ ).
2. The activation level of the activated document (ACT) is determined as a function of NACT and the current activation level (CACT). The two functions used here are SUM and MAX.

As an example, if a document with  $CACT = 0.3$  is activated by a nearest neighbor link from a document with score  $SACT = 0.4$ , with  $\alpha_{nn} = 0.7$  and the combining function SUM, the activation level ACT is

$$SUM(0.7 \times 0.4, 0.3) = 0.58$$

Table 2 shows the results of this experiment. The results are shown as precision figures after retrieving 5, 10 and 20 documents. The figures at the 20 document level are essentially identical because the search, as it is currently implemented, will effectively only reorder those 20 documents retrieved by the initial search. A better implementation would be to use the top 50 documents (say) in the initial ranking as starting points for the spreading activation search. In this way, some documents with initial ranks lower than 20 may be promoted into this top group. The main result shown in this table is that significant performance benefits were obtained (15% better than tf.idf at 5 document level, 11% at 10). The least successful search ( $\alpha_{nn} = \alpha_{cite} = 1$ ) is equivalent to retrieving each document in the top ranked probabilistic list along with its nearest neighbors and citations. The most successful search used low weights ( $\alpha_{nn} = 0.1, \alpha_{cite} = 0.2$ ) and the SUM combining function. Higher weights appear to give too much importance to the evidence provided by these links, and the MAX function, in most cases, prevents the alternative evidence from making any contribution to a document's score. Note that giving the activation links low weight does not mean that these inferences have low credibility.

Search	Precision at 5	10	20
Probabilistic (tf.idf)	0.45	0.36	0.29
$\alpha_{nn} = \alpha_{cite} = 1$ , MAX	0.34	0.33	0.25
$\alpha_{nn} = \alpha_{cite} = 0.7$ , MAX	0.49	0.37	0.28
$\alpha_{nn} = \alpha_{cite} = 0.7$ , SUM	0.48	0.38	0.28
$\alpha_{nn} = \alpha_{cite} = 0.1$ , SUM	0.50	0.40	0.29
$\alpha_{nn} = 0.1, \alpha_{cite} = 0.2$ , SUM	0.53	0.41	0.29

Table 2: Results from simple spreading activation search

Rather, it means that this additional evidence is less important than the evidence provided by the initial search.

Another series of experiments were performed that allowed activation to spread along nearest neighbor links (up to 10 links). The results obtained were virtually identical to those obtained using the single link activation. One reason for this was that the network of nearest neighbor links is very sparse and tends to form small, highly-connected clusters. Another reason is that the activation level that is propagated tends to drop off very quickly and therefore later activations do not change a document's score significantly.

#### 14.3.1.4 Further Experiments

The CACM document collection is ideal for testing the spreading activation model since it includes citation data as well as nearest neighbor information. This is not generally the case with test collections and so further testing was restricted to five document test collections for which only nearest neighbor information is available.

There has been considerable interest in the use of nearest neighbors as a form of simple cluster [22,24,28] and, as noted above, these form the basis for the cluster searches in I<sup>3</sup>R. Griffiths *et al.* suggested that acceptable levels of retrieval effectiveness could be obtained by using a very simple classification, containing just the *nearest neighbor clusters* for a collection, where a nearest neighbor cluster contains a document and that document to which it is most similar [24]. They demonstrated that such cluster searches gave retrieval performance at least comparable with, and sometimes superior to, that for non-cluster searches over a wide range of test collections. It is therefore of interest to investigate whether this information can be used more successfully by means of the spreading activation model.

The test collections used in the experiments reported here are as follows:

- **Keen:** A set of document titles, augmented by manually assigned indexing terms, and queries on the subject of librarianship and information science.
- **Cranfield:** A set of documents and queries, characterized by lists of manually assigned indexing terms, on the subject of aerodynamics.
- **Evans:** A set of automatically indexed document titles and queries from the INSPEC database.
- **Harding:** A set of automatically indexed document titles and abstracts and queries from the INSPEC database.
- **LISA:** A set of document titles and abstracts and queries from the Library and Information Science Abstracts database.

Constrained spreading activation was tested using all five test collections with only nearest neighbor information. All documents were used as starting points rather than just the top twenty ranked documents from the probabilistic search, and all queries were included in the analysis. The probabilistic weights were calculated as inverse document frequency (idf) weights (since term significance weights were not available for any of the five test collections). The combining function used for calculation of the adjusted activation levels was the SUM function. The details of the results are given in [5].

The results of these experiments indicate that using only nearest neighbor information that it is possible to achieve improvements in precision for all the collections tested using spreading activation methods. Thus, with the five collections used here it has been possible to achieve consistent performance increases, whereas this was not possible with the simple nearest neighbor clusters which have been advocated by Griffiths *et al.* [24] and by Willett [28] as hereto the best way of using interdocument similarity information.

### 14.3.2 A Retrieval Model for Complex Documents

In this section we describe progress in developing a retrieval model based on plausible inference. In this discussion, we will focus on the retrieval of complex documents, which are multimedia objects with complex structure (such as this report).

#### 14.3.2.1 The Basic Model

Given a query of the form  $\{d|P(d)\}$ , the formula  $P(d)$  is a Boolean combination of propositions involving the attributes of a complex document type. The truth or falsity of these propositions for a particular document can often be determined by simple lookup in the database. Examples of this are propositions such as *author* = 'R.Krovetz' and *year* > 85. We refer to this class of propositions as *exact match*. In some cases, inference will be needed to determine which documents satisfy a query proposition. This is the basis of deductive database systems [30]. In realistic systems, however, deductive inference is not sufficient for effective retrieval. This is due to uncertainty in the facts stored in the database, the inference rules, and the query itself. The classic example of this need for plausible inference is with propositions involving text, such as *about*(caption, 'conceptual structure'). This proposition could be satisfied in varying degrees by figure captions that do not contain the exact string *conceptual structure*. We refer to propositions that involve plausible inference as *partial match*.

Although most document attributes could be involved in both exact and partial match propositions, we can make the following distinction based on typical propositions associated with attributes. Exact match attributes have well-defined domains and uncertainty for these attributes occurs in the specification of the desired values, rather than in showing that a document has these values. For example, a user who is uncertain of a particular date or author may give a range of values for these attributes, together with some information about which values are more likely. Partial match attributes, on the other hand, have domains with poorly defined semantics, such as text and graphics. Uncertainty for these attributes could occur in the facts, inference rules and the query.

In order to develop a retrieval model for complex documents, we shall first concentrate on showing how plausible inference can be formalized for the typical partial match attribute, which is text. In other words, we will initially assume that documents contain just text. We will then show how this model can be extended to include other attributes, including exact match attributes.

A number of approaches to plausible inference have been proposed. These include heuristic certainty measures, non-standard logics, Bayesian inference net-

works, the Dempster-Shafer model of evidential reasoning, and symbolic representations [15,13,11,3]. Researchers in information retrieval have had considerable success with simple probabilistic models [12,1] and, together with the similarity of the Bayesian networks to the representations used in some advanced IR systems [4], this led us to choose this approach.

A Bayesian inference network is a directed, acyclic dependency graph in which nodes represent propositional variables or constants and arcs represent dependence relations between propositions (see [11] for a complete treatment of Bayesian nets). If a proposition represented by a node  $p$  directly implies the proposition represented by node  $q$ , we draw a directed arc from  $p$  to  $q$ . If-then rules in Bayesian nets are interpreted as conditional probabilities, that is, a rule  $A \rightarrow B$  is interpreted as a probability  $P(B|A)$ . The arc connecting  $A$  with  $B$  is labelled with a matrix that specifies  $P(B|A)$  for all possible combinations of values of the two nodes. The set of matrices labelling arcs pointing to a node characterize the dependence relationship between that node and the nodes representing propositions naming it as a consequence. For the class of rule bases we are interested in, these link matrices are fixed by the rule base and do not change during query processing. Given a set of prior probabilities for the roots of the DAG, these compiled networks can be used to compute the probability or degree of belief associated with the remaining nodes.

Different restrictions on the topology of the network and assumptions about the way in which the connected nodes interact lead to different schemes for combining probabilities. In general, these schemes have two components which operate independently: a predictive component in which parent nodes provide support for their children (the degree to which we believe a consequent depends on the degree to which we believe the antecedents of the rules naming it), and a diagnostic component in which children provide support for their parents (if our belief in a proposition increases or decreases, so does our belief in the antecedents that name the proposition as a consequent). The propagation of probabilities through the net can be done using information passed between adjacent nodes.

To see how a Bayesian network relates to the underlying probabilistic model, consider the simple network shown in Figure 1. In this network, the nodes labelled D1 to D4 represent propositions that a document has a particular content. The specific content is that of the corresponding document. The nodes labelled C1 to C7 represent propositions that a document is represented (indexed) by a particular concept. The directed links from a D node to a C node represent the conditional probability  $P(C|D)$ , which is the probability that a particular concept represents a document given that it has a specific content.

The query node Q represents a conjunction of C propositions. During re-



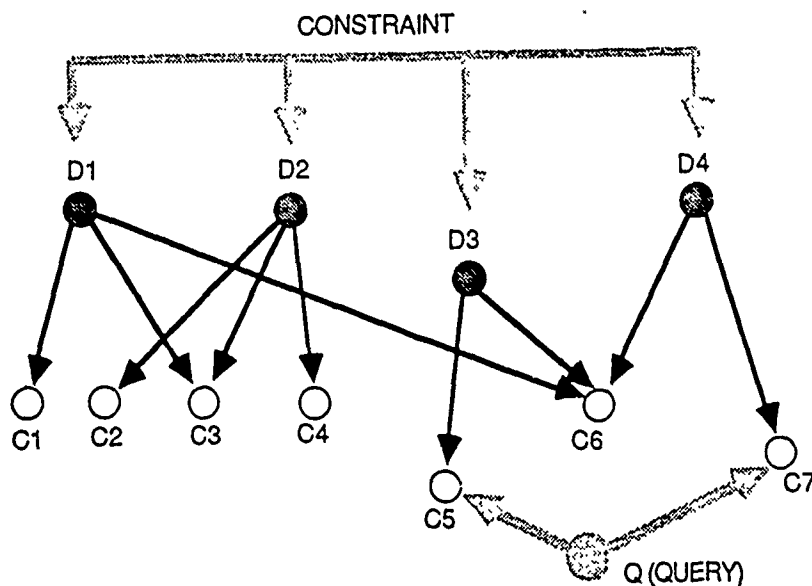


Figure 1: A Simple Bayesian Network

trieval, a constraint is applied to the network that restricts a single D node to be initially true. The network is then used to compute the probability of Q being true given this particular content (or document). This constraint is applied to each D node in turn (conceptually, at least) and the end result is a ranking of documents according to the values of  $P(Q)$ . Note that retrieval, as in a deductive database, is implemented as a form of inference. In the particular forms of Bayesian inference networks used here, we could in fact regard  $P(Q)$  as the probability that Q could be proved true.

We can also describe this use of the network in terms of an explicit probabilistic model. It can be shown that, given certain assumptions, the optimum retrieval effectiveness will be obtained by ranking documents according to estimates of  $P(Q|D)$ , which is the probability that the query is true given a particular document [32]. This *Probability Ranking Principle* is usually expressed in terms of probability of *relevance*, but we believe that our formulation has some advantages. The probability  $P(Q|D)$  can be transformed into an expression that involves the indexing concepts using standard probability theory as follows:

If we use  $R_i = (C_1, C_2, \dots, C_m)$  to refer to a particular indexed representation of a node, or in other words an assignment of concepts to a node ( $C_i$  is true when assigned), then

$$P(Q|D) = \sum_i P(Q|R_i, D)P(R_i|D)$$

and since  $Q$  and  $D$  are conditionally independent with respect to  $R$ ,

$$P(Q|D) = \sum_i P(Q|R_i)P(R_i|D)$$

Now we apply Bayes' theorem,

$$P(Q|D) = \sum_i \frac{P(R_i|Q)P(Q)}{P(R_i)} P(R_i|D)$$

At this point, we can use various approximations to estimate these probabilities. The most common assumption is independence, e.g.

$$P(R_i|Q) = \prod_{j=1}^m P(C_j|Q)$$

although we shall see that the inference network can represent dependencies between concepts. Given the independence assumptions, we get

$$P(Q|D) = P(Q) \prod_{j=1}^m \left( \frac{P(C_j|Q)}{P(C_j)} P(C_j|D) + \frac{1 - P(C_j|Q)}{1 - P(C_j)} (1 - P(C_j|D)) \right)$$

This expression is the basis of the network shown in Figure 1.

As more complicated models are introduced, the inference network becomes a more convenient representation for calculation of the probabilities. In Figure 2, we show a simple Bayesian inference network extended to include dependencies between documents and between concepts. Probabilities of the form  $P(C_j|C_k)$ , which capture thesaurus and statistical relationships between concepts, allow us to calculate  $P(Q|D)$  more accurately than the simple model assuming independence. Similarly, information about relationships between documents derived by clustering algorithms or from hypertext links is captured by probabilities of the form  $P(D_j|D_k)$ . We have introduced *evidence* nodes (those labelled  $e$ ) in this network to show that the dependencies between concepts and documents are of the form  $P(C_j|C_k, e)$ . The evidence is the facts stored in the database, such as `is_a(Concept1, Concept2)`. When several pieces of evidence relate to a conclusion, such as multiple links between concepts, a model of disjunctive interaction [11] can be used.

Another extension that can be made to the network is to allow uncertain queries. The easiest way to understand this is to imagine that users express their information needs in the form of text. It will then be possible to derive a range of queries from this text in the same way that different indexed representations of documents can be derived from their contents. The network can then be regarded

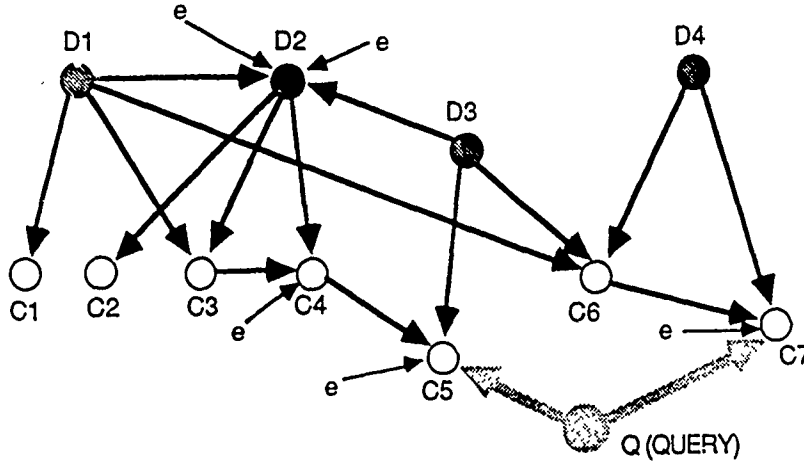


Figure 2: Bayesian Inference Network Incorporating Node and Concept Dependencies.

as computing  $P(I|D)$ , or the probability that a textual information need is satisfied given a particular document. More specifically, we can develop the extended model as follows:

Starting with the expression derived previously and replacing  $Q$  with  $I$  we get

$$P(I|D) = \sum_i \frac{P(R_i|I)P(I)}{P(R_i)} P(R_i|D)$$

If we now include the intermediate variable  $Q$  as was done for the document representations, the expression becomes

$$P(I|D) = \sum_i \left( \frac{P(I)}{P(R_i)} P(R_i|D) \sum_j P(R_i|Q_j) P(Q_j|I) \right)$$

The probabilities could then be estimated using independence assumptions as before.

The practical use of this extension is to allow users to specify which parts of their query are more important, as was done in the I<sup>3</sup>R and VAGUE systems [4,31]. Our interpretation of this process is that users are estimating  $P(Q_j|I)$ .

#### 14.3.2.2 Complex Document Retrieval

The model presented in the previous section enables us to retrieve a ranked list of text documents in response to a user's query. Multiple sources of evidence about the text content can be used in determining a document's position in the ranking. In order to handle the type of query described in the first section, two major features of these queries must be considered. First, these queries consist of a number of propositions of different types (exact and partial match) and second, the propositions involve specific attributes of a document (i.e. specific parts of the logical structure).

The first of these features fits easily into the Bayesian network framework. Multiple propositions are easily accommodated and, in fact, are used in queries in the text retrieval model in the previous section. The probabilities used in the text model are estimated either from the statistics of the document texts in the current database or from user estimates, as in the case of the queries. The probabilities associated with other partial match attributes could be acquired from users. This is the situation in the VAGUE system, where users are asked to estimate the similarity of domain values. Exact match propositions are easier in the sense that the probabilities associated with them are either 1 or 0 and are determined by lookup (or perhaps deductive inference). Note that there will be probabilities associated with exact match attributes in queries, however.

The main issue in dealing with propositions about specific attributes of a document is in constructing the inference network for a particular query. In order for this retrieval model to be practical, it is important that networks can be put together efficiently or that they already exist. In the case of simple text documents, the network is "precompiled" for all queries. This cannot be done for more complex documents but constructing a network is straightforward. A complex query network will have one node for each exact match proposition and one instance of a precompiled network for each partial match proposition. For text attributes, the probabilities in the precompiled networks could either be estimated from the text that is used in a specific attribute in the database, or from all text in the database. In the first case, there will be a network for each text attribute whereas in the second there is only one precompiled network.

### 14.3.3 Natural Language Processing and Information Retrieval

Information retrieval (IR) is concerned with techniques for processing natural language texts in order to satisfy a need for information. We can categorize tasks fitting this description into: *classification* (sorting of texts into a fixed set of categories), *retrieval* (locating texts that are related to an arbitrary expression of interest), *extraction* (recognizing of information in text and representing of this information in a format with known semantics, such as relational database fields), *summarization* (producing a new textual representation with content related to that of the original), and *question-answering* (making use of an understanding of the text content in order to answer questions). Classification and, particularly, retrieval are the traditional tasks that IR researchers have investigated. Extraction, summarization, and question-answering have been the focus of natural language processing (NLP) researchers. We will argue, however, that techniques developed for solving these latter problems have great relevance to IR.

The input to an IR system, called a *query*, is usually a textual description of the content of documents the user would like to see, such as:

Measurement of plasma temperatures in arc discharge using shock wave techniques.

The system compares the query to a large database of texts and outputs a list of document titles, abstracts, or full texts that the system has determined to be related to the user's interest. Usually this list is ranked in order of presumed relevance to the user.

One way to evaluate the performance of such a system is in terms of *precision* (what proportion of documents retrieved are relevant) and *recall* (what proportion of relevant documents are retrieved). Given a test collection consisting of queries, documents, and a set of judgments as to which documents are relevant to which queries, a *recall-precision curve* can be generated from a ranked list of documents. Such a curve shows how a user can, for instance, achieve greater recall at the cost of lower precision, by looking farther down in the ranking of retrieved documents.

The central task of IR can be viewed as inference [13], in particular the task of inferring whether concepts referred to in a query are also present in a document. Rather than using explicit inference rules, IR systems typically rely on implicit inferences in the form of *keyword matching* techniques. These techniques compute some measure of the similarity of the words in a query to those appearing in each document. Documents are then ranked according to this measure. Usually only the *content words* in the query and document are used in computing similarity. The words ignored are mostly function words, such as prepositions and determiners,

which reflect little of the content of a particular document. The set of words, multi-word phrases, and other information that is chosen to represent the document in retrieval processing is called a *document representative*, and the process of producing such representatives (a subject of much research in IR) is called *indexing*.

The most effective of the keyword matching techniques is probabilistic retrieval [12]. In probabilistic techniques, the value assigned to a match between a document word and a query word depends on the word's statistical distribution both within a document and within a collection of documents. Like all keyword techniques, probabilistic retrieval makes relatively little use of the structure of human language. Put another way, probabilistic retrieval could be applied, essentially without change, to recognizing similarities between computer programs, road maps, or strands of DNA.

How can IR be improved by using knowledge of language? Consider one technique already used in IR systems: *stemming*. A stemming algorithm uses morphological information to remove suffixes from words, thus reducing variations on a root word to a single form or *stem*. Suppose we view keyword matching on words as making implicit use of the inferences "All instances of a word refer to related concepts" and "Any two different words refer to different concepts." Then keyword matching on word stems uses the inferences "All instances of a word with the same root word refer to related concepts" and "Any two words with different root words refer to different concepts." The fact that stemming improves performance somewhat means that the latter inferences are more accurate, though they are still very fallible.

Stemming produces a *representation* of text which makes explicit one aspect of the morphological structure of words, and this allows more accurate inferences to be made. In Section 14.3.3.1 of this paper, we survey a variety of techniques, some using NLP and some not, that produce representations of text. The NLP techniques range from the use of simple syntactic templates for selecting indexing phrases, to the use of natural language *parsers* producing detailed syntactic and semantic representations. The non-NLP techniques involve various sorts of content analysis by trained or untrained humans. The common feature of all the above techniques is that they produce representations that make explicit some information about the original text structure, thus allowing more powerful inferences.

The conclusions we draw from our survey of techniques are presented in Section 14.3.3.2. To summarize, we find that language-oriented techniques have so far achieved minimal success in improving IR performance. However, we believe that NLP and knowledge representation (KR) techniques originally developed for the more difficult tasks of extraction and question answering can also be applied

to retrieval. In particular, the use of natural language parsers producing symbolic representations of text content will allow much more accurate inferences to be drawn by IR systems. In Section 14.3.3.3 we discuss the architecture of the ADRENAL testbed, which we are constructing to test specific hypotheses arising from these conclusions. In its final form, ADRENAL will allow the combining of traditional IR techniques with text skimming and parsing, and will take advantage of a core technical knowledge base and lexicon, of a large machine-readable dictionary, and of knowledge bases and thesauri for particular technical domains.

Constructing such a system will be a major undertaking, calling for some caution. Other reasons to proceed slowly in language- and knowledge-based IR are questions of efficiency, domain coverage, and the choice of appropriate KR methods. For this reason our initial work has stressed the use of hand-coded text representations and pre-existing parsing and KR tools. Experiments applying these to established test collections help unearth important issues that must be dealt with in a full-scale system, and provide some preliminary information about the performance improvements that can be expected.

#### **14.3.3.1 Previous Research**

In the following section we divide research on applying knowledge of language to IR into four groups. The first two groups concern NLP work on making explicit the syntactic or semantic structure of real-world text. We concentrate on techniques developed specifically for retrieval, but also examine other NLP research that has addressed real-world texts. A third group of work makes use of weak, nonstandard, language analysis methods for concept recognition, combined with a strong emphasis on domain knowledge bases. Finally, we look at work on manual methods for representing text content for retrieval, both those that use traditional indexing languages, and those using KR languages.

##### **14.3.3.1.1 Syntactic Analysis**

Many researchers have attempted to make use of syntactic information in IR. The omission of function words (e.g. prepositions, determiners, pronouns, etc.) when comparing documents and queries is a simple example of this. Slightly more complex are methods that use simple syntactic templates to recognize groups of words (*phrases*) on which to index a text. An example of such work is the FASIT system [35]. FASIT first tags words with their syntactic classes by using an on-line dictionary. It then indexes the text on those groups of consecutive words that fit one of several templates, such as <JJ NN> (adjective followed by noun). Experiments

with a small test collection showed no improvement over a conventional statistical retrieval system, but FASIT's designers felt their technique had more potential for improvement than traditional methods.

Other systems using syntactic templates to choose indexing phrases are LEADER [36], AIR/PHYS [37], and the TMC Indexer [38]. Some researchers have even gone beyond templates to use a full natural language parser in selecting indexing phrases [39,40]. It has not been demonstrated, however, that any of these syntactic indexing techniques significantly improve retrieval performance. For instance, Fuhr and Knorz present results for 300 queries and 15,000 documents showing that AIR/PHYS gives results similar to that of manual indexing. This is not particularly encouraging, as we will see in Section 14.3.3.1.4.

The above work implicitly assumes that phrases with certain syntactic patterns refer to meaningful concepts, so that when such a phrase appears in both query and document, the two refer to some common concept. These schemes are limited by the fact that the phrase must appear in the same form in document and query in order for the concept to be matched. Since the same concept can be expressed by many different syntactic structures (e.g. "information retrieval", "retrieval of information"), these methods miss many matches. However, if a system has knowledge about what syntactic relationships express similar semantic relationships, then it can recognize some superficially different occurrences of the same concept. In practice this knowledge is usually used first to select which phrases from text are meaningful, and then to perform *normalization* so that related phrases are all mapped to a single form.

A variant on normalization is to take information about what words in a query are related syntactically, and infer from this that occurrences of the words will be statistically dependent in relevant documents. This information can then be used to improve the performance of statistical retrieval [41,42].

The most thorough examination of phrasal techniques was done by Fagan [43]. Fagan used both statistical and syntactic methods to produce normalized phrases for documents from several test collections. He found that any sort of phrasal indexing made a relatively small improvement in retrieval performance. His best results were in the range of a 20-25% improvement in average precision, but results were much worse on some test collections. Syntactic means of selecting phrases performed no better than statistical methods, though Fagan points out that there are many ways that his syntactic techniques could be improved. Our conclusion from examining the work by Fagan and others [44,45,46,47,48], is that purely syntactic text representations have limited potential for improving retrieval.



#### 14.3.3.1.2 Semantic Analysis

The efficacy of syntactic normalization is limited by the fact that the same words must appear in both query and document to allow a match. To overcome this limitation, some use must be made of the meaning of the words. In IR, such semantic knowledge is usually introduced in the form of a *thesaurus*, which lists words related by synonymy, generalization, and similar relationships. This method goes beyond stemming to allow matches on words which have related meanings but unrelated surface forms. Thesauri have been shown to have some effectiveness in improving retrieval [49,50] and the availability of large on-line thesauri [51] may increase the usefulness of this technique. A few studies, such as Salton and Lesk [49] and Fuhr and Knorz [37] have looked at combining thesauri with syntactic normalization, but resulting improvements have been marginal.

NLP approaches semantics from a different direction. Rather than representing meaning by some combination of the text's original words, NLP systems usually represent the text's meaning by data structures built using a knowledge representation language. Meaning is expressed in these languages in terms of categories, instances of categories, and relations between those instances and categories. A data structure commonly used by KR languages is the *frame*, a cluster of relations associated with a category or category instance. If we consider frames and other knowledge structures as playing the role in NLP that keywords do in IR, then *knowledge bases* play the role of a thesaurus, by representing the relationships between categories, category instances, and relations. Knowledge base designers typically use a much richer set of relationships between concepts than are used in thesauri. KR languages ease this task by providing support for inheritance, inference rules, error-checking, and other features. Brachman and Levesque [52] is a good introduction to KR languages.

Representations such as frames provide a sort of semantic normalization by mapping linguistic structures with similar meanings to similar representations. By reducing the ambiguity and sheer number of entities that must be referred to, these representations make it more practical to write inference rules which recognize connections between non-identical concepts and which infer the presence of unmentioned concepts. A rare example of applying NLP-produced semantic representations to text retrieval is a paper by Sparck Jones and Tait [53]. This work used a syntactic parser, aided by semantic expectations, to produce a set of possible semantic frames for a query. Each frame was used to produce a number of rephrasings of concepts from the query, and all these phrases were combined into a single request to a retrieval system. This approach is similar to syntactic normalization, differing in that the semantic representation potentially generates a broader range

of rephrasings. The results were encouraging, but based on too small a testset to allow conclusions to be drawn about the usefulness of the method.

The Sparck Jones and Tait work is worth noting for two reasons. First, unlike much work in NLP, their parser made use of a knowledge base designed for general text processing rather than one tailored to the particular domain (physics and electronics) of their texts. This is important since it suggests that large domain-specific knowledge bases are not necessarily needed to support a useful degree of parsing of technical prose. Secondly, this is possibly the only work on parsing into semantic representations which was oriented strictly toward retrieval rather than toward more complex tasks such as extraction, summarization, or question-answering. These latter tasks are less tolerant of inaccuracy and uncertainty than retrieval, forcing researchers to concentrate on narrow domains and unrealistically simple text, thus making it difficult to evaluate the usefulness of the techniques to retrieval.

Still, any work on the mapping of text into a form which abstracts away from syntactic structure and word choice, and which makes explicit some of the text's meaning, has great potential to aid retrieval. Work on TOPIC [54,55] has devoted some attention toward parsing for retrieval, though that is not the only focus of the work. The TOPIC parser makes use of expert procedures for handling particular syntactic structures, aided by semantic expectations from a 150 frame knowledge base on computer technology. The output is a set of frames which are linked to each other and to the original text passages that produced them. This representation is used for both summarization and retrieval. The NLP techniques used by TOPIC are relatively sophisticated, and the text handled is realistic and broad (magazine articles on computer technology), but unfortunately no empirical studies measuring the retrieval performance of the system have yet appeared.

Most other work on analyzing real world texts has focused on extraction or question-answering rather than retrieval, and has usually involved text from narrow domains, such as naval ship-to-shore messages [56], wild flower descriptions [57], medical reports [58], banking telexes [59,60], chemical reaction descriptions [61], news stories on corporate takeovers [62], and police reports [63]. A few have handled broader domains such as technical patent abstracts [64] or newswire stories [65]. Some of these systems have achieved considerable accuracy in their extraction of information, suggesting that if they were combined with inference rules making use of the extracted information for comparing queries to documents, their performance as retrieval systems might be impressive. One attempt to do this is FERRET [66], which uses the newswire story skimmer FRUMP [65] to analyze computer network messages on a variety of topics. However, no performance data on FERRET have

yet been published.

#### 14.3.3.1.3 Knowledge-Based Concept Recognition

In the techniques discussed so far, knowledge about language, and automatic recognition of language structure, are important elements in recognizing when two pieces of text refer to related concepts. Other methods, however, put their emphasis on the reasoning that is done once a symbolic representation of a concept is obtained, while using language-weak methods to perform concept recognition. Recognition is usually done by a large number of rules, each of which infers the presence of some concept based on the occurrence of word *patterns* in text. These patterns combine sets of related words (such as a thesaurus would have) with simple syntactic templates or specifications that certain words occur within the same sentence, paragraph, or other context. Such rules are less accurate than a full NLP parser, but they do not fail on difficult linguistic constructions, as many parsers do.

The prototypical example of such a system is RUBRIC [67,68]. Users of RUBRIC can define rules for recognizing concepts from word patterns or from the presence of other concepts. One important point made by Tong, et al. [67] about RUBRIC was that allowing rules to refer to other concepts, as well as to text patterns, made them easier to extend and modify. Traditional and nontraditional logical operators can be used in the rules, and research on RUBRIC has emphasized experimentation with inference and scoring methods to manage uncertainty in concept recognition. Very good retrieval performance has been reported on a collection of 730 newspaper stories, but no comparison was done with traditional methods [68]. This makes it hard to objectively evaluate RUBRIC's performance, since retrieval results vary widely between test collections.

Another rule-based retriever is the news story classification system reported by Hayes, et al. [69] This system makes use of more sophisticated linguistic patterns than does RUBRIC, but uses much simpler scoring methods. Its rules are essentially arbitrary LISP code, and must be written by programmers rather than users, so the system can recognize only a small set of categories. A test where 500 newswire stories were classified by the system with respect to 6 categories showed recall and precision of 93%—similar to that of the human categorizers the system was compared to.

Another use that has been made of pattern-matching concept identification rules is to perform an initial classification of a text, so that the correct set of category-specific NLP techniques can be used to extract information from it. Two examples of systems using this technique are FRUMP [65] and TESS [59].

#### 14.3.3.1.4 Knowledge Representation without NLP

Even rules that recognize concepts in terms of word patterns can be complex to produce, so there is a long tradition of IR work which attempts to gain the benefits of associating text with semantic information without using any automated language analysis. As mentioned earlier, one method of doing this is to use a thesaurus, which enables non-identical words with related meanings to be matched. A more complex means of associating semantics with words is *manual indexing* using a *controlled vocabulary*. In these methods a human indexer produces a document representative for each piece of text. The representative is a set of *index terms* (words and phrases), but semantic normalization is provided by requiring that all index terms be drawn from a fixed list. This contrasts with *full text indexing* which, since it uses all content words from text, is easy to automate but does not accomplish any semantic normalization. The more complex manual indexing schemes require the indexer to indicate relationships between the terms as well. The relationships may be represented as explicit symbols (similar to slots or links in KR languages) or they may be implicitly encoded by requiring the indexer to first list a general subject class for the text, and then follow it by more specialized terms.

A number of studies have cast doubt on the efficacy of manual indexing. One early study which carefully evaluated manual indexing schemes was by Aitchison and Cleverdon [71]. This work compared the indexing and retrieval of metallurgical documents using the WRU Index Language for Metallurgy with the more general English Electric Faceted Classification for Engineering. The results, based on approximately 100 queries and 1000 documents, showed no advantage for the more domain-specific scheme. This argues against manual indexing as an effective means of using domain knowledge. Furthermore, the absolute performance figures for both systems (approximately 80% recall with 10% precision) were similar to those achieved by full text indexing.

Aitchison and Cleverdon report that many of the failures to retrieve relevant documents resulted from the human indexer's omission of important concepts, or from the system's failure to recognize matches between related concepts. Retrieval of non-relevant documents mostly resulted from matches on concepts that played only a minor role in a document, and from matches on query concepts that were expressed in too general terms.

Questions about the efficacy of manual indexing schemes have not stopped their development. Some examples include SYNTOL [72], PRECIS [73], MeSH [74] and Relational Indexing [75,76]. Work on Relational Indexing even explored the use of inference rules to infer connections not explicitly represented in text [77], a technique usually found only in sophisticated NLP systems. Unfortunately, no

performance evaluation was ever published on this feature.

Recently, work has been done in the AI community on representing text content for retrieval, question answering, and other tasks. These methods use KR languages rather than indexing vocabularies, and allow a much richer representation of the concepts in text. Applications include medical case reports [78], historical information [79], an artificial intelligence reference book [80], and text from Chemical Abstracts [81].

Most of these systems have not been evaluated under controlled conditions. An exception is GRANT [3], which compares frame-based representations of research proposals and funding agencies in order to determine what agencies are likely to fund a proposal. The degree of match between two frames is based on heuristics which makes use of a large semantic net. GRANT achieved a recall rate of 67% at 29% precision (based on experts' judgments of what the appropriate funding agencies for a proposal were). This outperformed the simple keyword-based system it was compared against, and is about the same performance as the best probabilistic keyword methods. Other encouraging results come from D. E. Lewis' dissertation, where a high correlation was shown between relevance of documents and partial matches on case frame representations of queries and document abstracts [82].

#### 14.3.3.2 Lessons from Previous Research

Our survey provides both discouraging and encouraging evidence for the potential of language-oriented retrieval. On the minus side, NLP techniques for building semantic representations require large amounts of knowledge and have only been demonstrated in narrow domains. More broadly applicable techniques, such as syntactic normalization, provide relatively small performance improvements. Techniques such as concept recognition rules and controlled vocabulary indexing take considerable human effort and achieve uncertain performance. Little of the work on the more advanced language- and knowledge-oriented techniques has used test collections or been compared with established IR techniques, making competing claims hard to sort out. However, we believe the following positive results have also been established:

- *Representations which abstract away from surface text can improve retrieval performance by making it easier to implement effective inferences.* The traditional IR inference technique—counting of matches between identical items—can usually be improved by normalization techniques, such as stemming, thesauri, and phrasal indexing, although the degree of improvement varies widely between techniques and between different retrieval situations. Work by Tong,

Cohen, and others points toward the ability to state simple but powerful inference rules given a representation of text's semantic content. Note that one might take the failures of manual indexing to be an argument against semantic representations, since controlled vocabularies provide some of the same features as KR languages. However, since many problems with manual indexing stem from human inconsistencies, we instead conclude, as did Carbonell, et al. [83], that content analysis should be automated, and that richer semantic representations and inferences should be used.

- *Frames are a particularly good representation of document content for retrieval.* Manual indexing usually requires exact matches between indexing terms. Therefore, relevant documents expressing related, but not identical, concepts are missed. Many nonrelevant documents are retrieved, because they contain concepts which occur in the query but do not occur in the proper relationships with other concepts. By explicitly representing relationships between concepts as slot relations, and then grouping these slots in a frame for a concept, it is possible to allow partial matching, as well as matching which takes into account relationships between concepts. Early results on retrieval using partial matching of frame-based text representations have been encouraging [3,82]. Since these experiments made use of matches on only one slot per frame, there is potential (as Cohen and Kjeldsen concluded from their failure analysis) for considerably improving these methods by allowing matches on more than one relation at a time.
- *Current NLP techniques can provide useful analyses of real-world text.* Given some tolerance for error and ambiguity, meaningful semantic analysis can be done in constrained domains, while syntactic analysis is possible in any domain. The work by Sparck Jones and Tait suggests that even imperfect and ambiguous semantic representations have the potential to aid retrieval. The perception that current NLP techniques will necessarily fail on real-world text may apply only to more difficult tasks such as question-answering, and not to text retrieval.
- *The application of NLP to text retrieval is relatively unexplored in comparison with its application to other tasks.* Classification systems such as the one reported by Hayes, et al. use weak NLP techniques and achieve high performance on a relatively easy task. Text extraction systems achieve good performance on a very challenging task by using powerful NLP techniques in constrained domains. But little is known about the potential of powerful NLP techniques to improve text retrieval in broad domains, a task of intermediate difficulty.

### 14.3.3.3 ADRENAL

The ADRENAL testbed is our attempt to apply the lessons of the previous section to the retrieval of a wide range of scientific and technical documents. In this system, natural language parser produces a frame-based representation of the user query. ADRENAL uses this representation in generating a keyword query to a traditional retrieval system. The top ranked documents from the keyword-based retrieval are shown to the user for feedback while also being given to a *text skimmer* [65]. The text skimmer uses fast NLP heuristics to scan for concepts from the query, and then reranks the retrieved documents, so that the worst can be discarded and the best shown to the user.

Documents which are not discarded, including those shown to the user, are passed on to a full NLP parser. The purpose of this parser is not to build a complete semantic representation of the remaining documents, but rather to interpret enough of each document to confirm or disconfirm the presence of query concepts. The parser will build frame representations for those sentences containing words that caused the document to score well, and a matching component will see to what extent concepts appear in the same relationships as in the user query.

During the retrieval process, decisions by the user as to which retrieved documents are relevant will result in changes in the concepts being sought. This in turn may result in additional keyword queries being made to the traditional retrieval system, as well as resulting in changes in the concepts sought by the text skimmer and parser. The process will continue until the user is satisfied with the documents retrieved, and the system will compile statistics on its performance for later analysis.

Queries and documents will be parsed into frames drawn from REST (Representation for Science and Technology), our representation system. In its final form, REST will consist of:

1. An inheritance hierarchy of frames representing events, objects, and relationships common to a range of scientific and technical fields.
2. A lexicon specifying mappings from a general scientific and technical vocabulary to the hierarchy of frames. Morphological, syntactic, and other information needed for parsing will be recorded.
3. A set of inference rules, which will be used to infer the presence of implicit concepts, as well as to determine when related concepts can be matched.

The output of our parser will be a *frame network* of REST frames linked by slot relations. One difference from usual NLP representations is that some of

the original words from text will be present in the representation, along with the semantic abstractions the parser builds. This is because REST defines only high-level scientific concepts, so words referring to domain-specific concepts must be kept to avoid losing information.

Our emphasis in the ADRENAL project is on adapting existing NLP techniques to the task of retrieval, rather than on developing new NLP techniques. In this way we hope to establish results that are more generally applicable than if we had relied on a specialized parser of our own design. We will also learn a great deal about what would be desirable in a specialized IR parser.

Our initial work was with the Transportable Understanding Mechanism Package (TRUMP) [84,85], developed by Paul Jacobs and his colleagues at General Electric Research & Development. TRUMP is a *chart parser* [86], which means that it uses a context-free grammar and has a tabular data structure, the "chart," for representing syntactic structures recognized during a parse. Such a parser, in combination with a bottom-up control strategy, is especially suited to dealing with real-world text, as its representation of partial constituents allows some information to be output even when a complete parse fails.

In addition to the basics of syntactic parsing, TRUMP provides a semantic interpretation mechanism which creates frame structures in the KR language KODIAK [87]. TRUMP also has a mechanism for ranking alternative parses by syntactic and semantic criteria, and has extensive support for phrasal lexicons [88]. Implementation of the REST representation system in KODIAK was straightforward.

We conducted a number of experiments using TRUMP to parse text from the NPL collection of physics and electronics abstracts, a standard document retrieval testset [89]. These experiments taught us a good deal about the good and bad points of TRUMP, and about the difficulties of parsing real-world text in general.

Some of these difficulties are well-acknowledged in the parsing literature. Syntactic structures are very complex and highly ambiguous. The vocabulary is huge and most words have multiple possible lexical categories. Others were more surprising, such as the relatively large number of typographical errors and proper names in our IR test collections.

As for TRUMP, we found that its strongest point was its representation of linguistic and semantic knowledge. In particular, each context-free rule in its grammar has one or more *linguistic relations* associated with it. Rules mapping from syntactic to semantic structures are then defined in terms of these linguistic relations. (A similar technique was used in the Mitre Corporation's KING KONG parser [90].) This additional layer of abstraction makes modifications to both the



grammar and the semantic mappings much easier. Furthermore, the linguistic relations are themselves part of a taxonomic knowledge base, which allows considerable economies in represented information about them, as well as allowing a very clean representation of phrasal items as specific instances of linguistic relations.

On other counts, however, TRUMP did not prove to be the best choice for our ongoing work. The main problem was the inability of the system to deal with truly ambiguous semantic interpretations. TRUMP handles semantic ambiguity by means of *concretion*, a process which gradually refines a general interpretation to a more specific one on the basis of contextual information. This works well for those cases, which NLP systems have traditionally found difficult to handle, where a word or phrase has a large number of closely related meanings. Unfortunately, this mechanism is not appropriate for handling the many words which have two or more very different meanings. Since our emphasis in producing text representations is to recognize broad semantic distinctions rather than subtle ones, TRUMP was not a good match to our needs.

In addition, the grammar supplied with TRUMP was not adequate for producing even partial representations of the scientific texts we were dealing with. This is neither a criticism, nor surprising, since TRUMP had not been used in our particular application before. Finally, with an eye toward the future, we were interested in obtaining a parser that could be more easily modified and redistributed to other researchers than the experimental and proprietary TRUMP software. (For another view of TRUMP, see the paper by Jacobs and Rau elsewhere in this issue.)

The second phase of our research on adapting parsing for IR is currently underway. It makes use of MCHART [91], a chart parser written by Henry Thompson of the University of Edinburgh. MCHART is less efficient and considerably less complex than TRUMP, does not come with its own large grammar, and does not provide mechanisms for semantic interpretation. However, it is easily modified, well-documented, and readily available. It also uses an agenda-based control mechanism [92] which allows great flexibility in experimenting with alternative parsing strategies, ranking of parses, and integrating different sources of knowledge in parsing [93,94].

Continuing with our strategy of making use of existing resources, we are currently experimenting with the Linguistic String Project (LSP) Grammar [58], a moderately large syntactic grammar that has been extensively used for the parsing of technical (especially medical) prose. This grammar has the advantage of being large and well-documented. Its main disadvantage is that it assumes the presence of approximately 200 extragrammatical *restrictions* [58], which are used during parsing to prune unpromising analyses. We are currently investigating what fraction of these

restrictions we need to implement in order to achieve reasonable performance for our task.

Our lexicon makes use of a lispified version, prepared at the University of Cambridge [95], of the on-line Longman Dictionary of Contemporary English (LDOCE). This dictionary provides syntactic class information on approximately 35,000 distinct words. In combination with a locally developed morphological analyzer which reduces inflected words to their root form, this gives us an effective vocabulary of around 100,000 words. LDOCE is not that big as dictionaries go, but our initial experiments with a small grammar show that the parser is usually able to disambiguate the remaining unknown words. We plan to increase the accuracy of this disambiguation by introducing an additional set of suffix-based heuristics [96] for inferring the syntactic category of unknown words, and possibly by making use of a stochastic tagging procedure [97,96].

In addition to the syntactic category information provided by the LDOCE, we are currently expanding our REST knowledge representation system to include detailed syntactic and semantic information on a scientific and technical vocabulary of around 1500 words. This information includes the interpretation rules which map from individual words to categories and from linguistic relations to semantic relations. Since we are making use of the strategy of associating semantic interpretation rules with linguistic relations, one of our modifications to the LSP grammar is to specify the linguistic relations associated with each context-free rule.

#### 14.3.3.4 Experiments With ADRENAL

We have carried out some initial experiments with ADRENAL and the associated REST language. The obvious one was to provide an initial test of our hypotheses that retrieval performance can be improved by using explicit semantic representations, and in particular by techniques that measure the similarity of frame-based representations of queries and documents. The benefits of semantic representation were demonstrated by the various ways in which REST representations were used to enhance tf.idf retrieval. Both textname matching and relation triple matching improved retrieval performance, showing the usefulness of ranking documents based on frame similarity measures. These improvements were demonstrated using representations of queries and documents similar to those we might expect to get from a natural language parser. Our experiments, however, are based on a quite small set of hand-coded documents, and can only be verified by using automated analysis of larger collections.

It must be strongly emphasized that these results do *not* in any sense provide an upper limit on the performance of language-oriented retrieval using document

analysis, just because the representations were hand-coded. The set of frame categories in REST is currently being greatly expanded beyond those used in these experiments, inference rules are being added, and the particular choices of slot types are being experimented with. A robust parser will be able to analyze many more documents per retrieval than we were able to analyze by hand. The above experiments did not use domain knowledge bases or domain lexicons, which will undoubtedly enhance performance, as will the use of full-text documents rather than abstracts. More sophisticated frame matching methods, already developed by AI researchers, need to be tried.

The second thing we wanted to do with these experiments was to demonstrate the strong influence that representation choices have on what retrieval methods are appropriate and on what performance they achieve. These choices range from whether documents are represented by abstracts or full text, to what parts of a text are translated into a KR language, to what slot and frame categories are used. What we found was that choices made in human coding of the documents strongly influenced the performance that particular matching techniques could achieve on the resulting frame network representations. These representation choices caused a number of apparently quite different matching techniques to give similar results, but also revealed differences between others. There was also evidence that anticipating our simple matching technique (relation triple matching) in selecting the parts of text to represent allowed an even simpler technique (keyword matching) to be improved.

The interactions between text analysis and matching techniques will not go away when a parser is used rather than a human coder. For the sake of efficiency it will be desirable to parse as little of a document as possible, and we will have to take into account the matching techniques used in deciding what parts of the text can safely be skipped. In addition to this explicit selectivity, any parser will implicitly introduce biases into the representations it builds, based on the parsing technique it uses, the coverage of its grammar, and the design of its knowledge representation language. These biases will interact with matching in unexpected ways, just as the explicit biases we gave our human coder did. These interactions are likely to be exacerbated with the use of inference techniques more complicated than matching, such as ones which infer unrepresented concepts.

There are two lessons to draw from this. The first is that the interaction of representation choices and parser strategies with matching and other inference methods needs to be considered early in the design of a system. The second lesson is that language-oriented retrieval makes the IR tradition of evaluating techniques on realistic testsets more important than ever.

#### 14.3.4 Structures for Plausible Inference in Semantic Networks

Can cough syrup make people drunk? Our favorite brand can, because it contains alcohol. If you didn't already know that cough syrup is intoxicating, you could infer it from two specific propositions—cough syrup contains alcohol and alcohol is intoxicating—and from a general plausible inference rule:

Rule 1	$x$	CONTAINS	$y$ , and
	$y$	CAUSES	$z$
	$x$	CAUSES	$z$

Other familiar rules of plausible inference include property inheritance (e.g., cats have five toes, Ginger is a cat, so Ginger has five toes) and causal abduction (e.g., fires cause smoke, so if you see smoke, look for a fire).

Rules like these have two roles that we expect to become increasingly important in coming years. First, they support *graceful degradation* of performance at the boundaries of our knowledge. A *brittle* knowledge system that doesn't know explicitly whether cough syrup makes you drunk won't offer a plausible answer—it simply won't answer the question [8,102,100]. Graceful degradation depends on general knowledge, which we formulate as plausible inference rules such as Rule 1, to make up for a lack of specific knowledge. Second, we expect plausible inference to reduce the effort of building knowledge bases, because knowledge engineers needn't state explicitly those propositions that can be plausibly inferred. Property inheritance, for example, relieves us from having to state explicitly that each member of a class has each property of that class [98]. Rules like property inheritance and Rule 1 obviously are needed to build "mega-frame" knowledge bases [102].

Rule 1 has the same structure as property inheritance over ISA links, and can serve the same purposes, that is, supporting graceful degradation and knowledge engineering. We have developed a simple method for deriving such rules from the relations in a knowledge base, and we have shown how to differentiate plausible ones from implausible ones based on their underlying "deep structure."

This paper describes two empirical studies of these rules. Both depend on a moderately large knowledge base that we developed for the GRANT project [2,3]. The GRANT KB contains roughly 4500 nodes linked by 9 relations and their inverses. In the first study we derived approximately 300 plausible inference rules from these relations. Then we generated over 3000 specific inferences by replacing the variables in the rules with concepts from the GRANT KB, and presented them to human subjects to discover which syntactically permissible rules were plausible (Sec. 14.3.4.1). The second study tested the hypothesis that the plausibility of these rules can be predicted by whether they obey a kind of transitivity (Sec. 14.3.4.2). We will begin

by describing these studies, hypotheses, and results. Then we will discuss the role of knowledge in assessing the plausibility of inferences.

#### 14.3.4.1 Experiment 1: Identifying Plausible Rules

In this section we describe how to use the structure of property inheritance to produce many other plausible inference rules, and how we determined the plausibility of these rules.

##### 14.3.4.1.1 Background

Property inheritance over ISA links can be written

$$\begin{array}{ccccc} n_1 & \text{ISA} & n_2, & \text{and} & \\ n_2 & \text{R} & n_3 & & \\ \hline n_1 & \text{R} & n_3 & & \end{array}$$

where the relation R between  $n_2$  and  $n_3$  is viewed as a property of  $n_2$ . For example, if a canary is a bird and bird HAS-COMPONENT wings, then canary HAS-COMPONENT wings. Here, R is HAS-COMPONENT and the inherited property is "HAS-COMPONENT wings." Many plausible inference rules have this structure, but inherit over links other than ISA. For example, in the "cough syrup" inference, above, cough syrup inherits the "CAUSES intoxication" property over the CONTAINS relation:

$$\begin{array}{ccccc} \text{cough syrup} & \text{HAS-COMPONENT} & \text{alcohol,} & \text{and} & \\ \text{alcohol} & \text{CAUSES} & \text{intoxication} & & \\ \hline \text{cough syrup} & \text{CAUSES} & \text{intoxication} & & \end{array}$$

Each pair of relations can produce two plausible inference rules that have the same structure as property inheritance over ISA links. For relations R1, R2 these rules are:

$$\begin{array}{ccccc} & n_1 & \text{R1} & n_2, & \text{and} \\ \text{Rule 2} & n_2 & \text{R2} & n_3 & \\ & \hline & n_1 & \text{R2} & n_3 & \end{array}$$

and

$$\begin{array}{ccccc} & n_3 & \text{R2-INV} & n_2, & \text{and} \\ \text{Rule 3} & n_2 & \text{R1-INV} & n_1 & \\ & \hline & n_3 & \text{R1-INV} & n_1 & \end{array}$$

Rules can be chained by letting the conclusion of one serve as a premise for another. Since each pair of relations produces two rules, a knowledge base constructed from  $N$  relations will produce  $(N^2 + N)/2$  pairs of relations (including relations paired with themselves) and an equal number of rules. The GRANT KB is constructed from nine relations and their inverses, so  $(18^2 + 18)/2 = 342$  were generated.

Experiment 1 had two goals. One was to generate all possible rules for the GRANT KB and to determine which of them produce plausible conclusions. The other was to find out how the plausibility of conclusions is affected by chaining these rules. Applying roughly 300 rules to the GRANT KB (as we describe below), produced thousands of first generation inferences and over 200,000 second-generation inferences. We expected very few of these to be plausible; but, if we could discover or predict the plausible ones, then we would have a powerful method to reduce the effort of constructing large knowledge bases.

#### 14.3.4.1.2 Design

To determine whether the rules produce plausible conclusions, we first *instantiate* them with specific concepts, then present them to human subjects to judge.

We derived 315 rules from the GRANT KB.<sup>1</sup> For each we produced 10 *test items* (five first generation items and five second generation items) by the following method:

Each rule is based on two relations. For each pair, say HAS-COMPONENT and MECHANISM-OF, we search the GRANT KB for triples of nodes  $n_1, n_2, n_3$  that are connected by these relations (i.e.,  $n_1$  is connected to  $n_2$  by HAS-COMPONENT, and  $n_2$  is connected to  $n_3$  by MECHANISM-OF). Each triple represents a pair of premises from which *two* inferences can be drawn (see Rules 2 and 3, above).

Most pairs of relations in the GRANT KB yield dozens of  $n_1, n_2, n_3$  triples. We randomly select five, and their conclusions, to be first generation test items. However, we add the conclusions of *all* the triples to the GRANT KB.

This procedure is repeated to generate second generation test items, with the added condition that one premise of each second generation item must be a conclusion that was produced during the previous search (though not necessarily the conclusion of a first generation test item).

In all, the 315 rules yield a data set of 3116 test items, of which roughly half are first generation and half are second generation items.<sup>2</sup>

---

<sup>1</sup>Pruning duplicates reduces the original 342 rules to 315.

<sup>2</sup>We don't have 3150 items because, for some rules, the GRANT KB yielded fewer than five first

#### 14.3.4.1.3 Procedure

Items in the data set were presented to human subjects by a computer program. Subjects were asked first to indicate whether both premises were acceptable, one or both were unacceptable, or they did not understand one or both premises. Next, the conclusion was shown and subjects were asked to judge whether it followed or did not follow from the premises, or else to indicate that they did not understand the conclusion. Each item was seen by two subjects. Following a practice session with 20 items (none of which was in the data set), each subject judged approximately 700 items from the data set. This took about five hours, distributed over three or four self-paced sessions.

#### 14.3.4.1.4 Results

Since the premises of the test items came from an existing knowledge base we expected that most would be judged acceptable. This is in fact the case: 82% percent of first generation premises and 63% of second generation premises were judged to be acceptable. The following results pertain only to those items.

Each rule is represented in the data set by five first generation items and five second generation items, and each item was seen by two subjects. Thus, 10 judgments are made of the items in each generation of each rule. Two *plausibility scores* for a rule, ranging from 0 to 10, are equal to the sum of the number of items that subjects judged plausible for each generation of each rule. The mean plausibility score, over the 315 rules, for first generation items is 4.18 (var. = 6.92), and the corresponding statistic for second generation items is 3.17 (var. = 4.88). Both are significantly different from chance and from each other at the  $p < .01$  level. The fact that both are *below* chance means that most rules are not plausible. Given this, one would expect chaining of inferences to produce increasingly-implausible conclusions. This is supported by the evidence that second generation inferences are significantly less plausible than first generation ones. Subjects judged approximately 50% of the rules to have plausibility scores between 3 and 7 (of a possible 10); they judged the rest of the rules to be predominantly plausible or implausible.

#### 14.3.4.1.5 Discussion

While these results indicate that many rules generate predominantly plausible conclusions, and many others are predominantly implausible, they do not tell us how to predict which will be plausible and which will not. We wanted to find  

---

generation instances.

a small set of common characteristics of rules on which to base these predictions. Furthermore, we wanted these characteristics to depend only on the relations in the rules, not on the nodes or any exogenous factors.

We discovered two common aspects of relations. Some relations, such as HAS-COMPONENT have a *hierarchical* interpretation. Others, such as CAUSES, can be interpreted as *temporal* relations. Lastly, relations such as MECHANISM-OF can have both hierarchical and temporal interpretations: in " $n_1$  MECHANISM-OF  $n_2$ ,"  $n_2$  may be a process that hierarchically subsumes the mechanism  $n_1$ , or  $n_1$  may be an object or process that exists or is required prior to achieving  $n_2$ . A *deep relation* has a h (hierarchical) or t (temporal) interpretation, or both. Expressing rules in terms of these deep relations reduces the set of 315 surface rules to 95 unique *deep structures*.

More importantly, we identified a characteristic of deep structures, called *transitivity*, that seemed to explain why some rules were plausible and others implausible. The mean plausibility score for transitive rules was 8.94 (out of 20; var. = 16.83), and for intransitive rules, 5.89 (var. = 14.46). Again, the preponderance of these rules are judged implausible, but these values are significantly different ( $p < .01$ ), and provide strong post-hoc evidence that transitivity is a factor.

Transitivity is clear when surface relations map to deep relations whose h and t elements point in just one direction. But the surface relations HAS-MECHANISM and PURPOSE-OF have deep relations where t and h point in opposite directions. Therefore, rules that are transitive under one interpretation of these relations are necessarily intransitive under the other.

Although our data suggested that transitivity predicts the plausibility of rules with unambiguous structures, the results were less clear for ambiguous ones. All ambiguous structures have transitive interpretations, but we knew from our data that not all the corresponding rules were plausible. We hypothesized a characteristic of interpretations, called *consistency*, that might discriminate plausible ambiguous rules from implausible ones. A structure has a consistent interpretation when its deep relations all have the same interpretation, either h or t.

#### 14.3.4.2 Experiment 2: Exploring Transitivity

At the end of Experiment 1, we had formed the hypotheses that transitivity predicts plausibility, and that consistency determines the interpretation (transitive or intransitive) of ambiguous structures. Experiment 2 tests these hypotheses.



#### 14.3.4.2.1 Design

Experiment 2 focused on ten relations from Experiment 1: CAUSES, COMPONENT-OF, MECHANISM-OF, PRODUCT-OF, PURPOSE-OF and their inverses. (The other relations replicate deep relations and occurred relatively infrequently in the knowledge base.) Since each of these surface relations has a unique corresponding deep relation, the 95 rules they generate map to 95 different deep structures. From these, we chose 56 structures (and thus, rules) as a representative sample.<sup>3</sup> We generated 10 first generation test items for each of the 56 rules, just as we did in Experiment 1.

#### 14.3.4.2.2 Procedure

Fourteen subjects each viewed all the test items. Items were presented as in Experiment 1.

#### 14.3.4.2.3 Results

Our hypothesis is that transitivity, as determined by the consistent interpretation of the deep structure, predicts plausibility. Eight rules are composed of surface relations that have just one deep interpretation (CAUSES, CAUSED-BY, HAS-COMPONENT, COMPONENT-OF. With these we can analyze the effects of transitivity and consistency on plausibility in rules with single interpretations. A two-way analysis of variance found a significant main effect of transitivity ( $p < .001$ ) and a significant transitivity  $\times$  consistency interaction ( $p < .001$ ), but no main effect of consistency ( $p > .2$ ), confirming that transitivity predicts the plausibility of these rules.

The results suggest that we cannot predict the plausibility of rules that have no consistent interpretation, because the mean plausibility score for these rules is roughly five out of 10 (i.e., at chance) irrespective of whether the rule is transitive.

Analyzing all our rules in terms of these categories yields 18 that have consistent transitive interpretations, 20 consistent intransitive rules, 8 inconsistent rules, and 4 rules that have both transitive and intransitive consistent interpretations.<sup>4</sup>

---

<sup>3</sup>Rules generated from a single surface relation and its inverse always map to one transitive and two intransitive deep structures. Our sample included the transitive structure and one of the intransitive structures (chosen randomly). Pairs of non-identical relations and their inverses form four transitive and four intransitive rules. Our sample included two transitive and two intransitive rules from each of these sets.

<sup>4</sup>Unfortunately, the test items for the other six rules shared many common premises. This was an unavoidable consequence of our decision to generate test items randomly. Four had consistent transitive interpretations, two had consistent intransitive interpretations.

#### 14.3.4.2.4 Discussion

While the predictive power of transitivity is high for rules that have only one interpretation, it becomes diluted in rules with multiple interpretations. It is not surprising that rules with consistent transitive *and* intransitive interpretations have a mean plausibility score roughly halfway between the scores for transitive and intransitive rules. However, the mean plausibility score of inconsistent rules, which we expected to be at chance, was higher (61%); and the mean plausibility score of rules with consistent intransitive interpretations, which we expected to be implausible, was not as low as we expected (43%).

We hypothesize that both these effects are due to an unanticipated factor that is raising the plausibility of some but not all of these rules. Whereas all our surface rules have the same structure as property inheritance over ISA links, some but not all of the *deep structures* of both the intransitive and inconsistent rules have this form. For example, the deep structure for the rule  $n_1$  COMPONENT-OF  $n_2$ ,  $n_2$  HAS-MECHANISM  $n_3 \rightarrow n_1$  HAS-MECHANISM  $n_3$  is intransitive, but its conclusion is often plausible.

*Generalized property inheritance* (GPI) is a characteristic of a rule's deep structure, comparable with transitivity:

If  $n_1$  is related to  $n_2$  by  $h$ , and  $n_2$  is related to  $n_3$  by any relation  $i$ , then it is plausible to infer that  $n_1$  is related to  $n_3$  by  $i$

This definition does not restrict the direction of  $h$ ; it can point "up" or "down" from  $n_1$  to  $n_2$ , whereas in property inheritance over ISA links,  $n_1$  must be a subclass or instance of  $n_2$ , that is, ISA must point "up." We relax this for GPI because it is often plausible to infer that a concept will have properties of those concepts hierarchically-inferior to it.

GPI explains why some intransitive rules have higher-than-expected plausibility scores. Since some transitive rules are also GPI, we ran a post-hoc transitivity  $\times$  GPI analysis of variance, and found main effects of transitivity ( $p < .001$ ) and GPI ( $p < .05$ ), with no interaction effect. Post-hoc tests on the means (Newman-Keuls) found a significant difference between GPI intransitive items and non-GPI intransitive items ( $p < .05$ ), which means that among intransitive rules, GPI differentiates two statistically-distinct classes—relatively plausible and relatively implausible rules. After removing GPI rules, the mean plausibility score of inconsistent rules decreases. Therefore, GPI provides a post-hoc explanation of why intransitive and inconsistent rules have higher-than-expected plausibility scores. Among transitive items, GPI had no statistically discernible effect. And since there was no interaction between transitivity and GPI, we regard them as independent factors.

### 14.3.4.3 Contributors to Plausibility

In this section we will discuss the factors that contribute to judgments of plausibility. (A more detailed analysis and presentation is given in [99].) Recall that our goal is to find plausible inference rules that support graceful degradation and help knowledge engineers. Ideally, the agent who uses these rules should not need much knowledge to judge the plausibility of their conclusions. For example, the plausibility of the conclusion of the rule

$$\frac{n_1 \text{ CAUSES } n_2, \text{ and } n_3 \text{ CONTAINS } n_1}{n_3 \text{ CAUSES } n_2}$$

seems not to depend on the objects that instantiate  $n_1$ ,  $n_2$ , and  $n_3$ . In contrast, to judge the plausibility of a conclusion of the rule

$$\frac{n_1 \text{ CAUSES } n_2, \text{ and } n_2}{n_1}$$

we need knowledge about  $n_1$  and  $n_2$  that can tell us how likely  $n_1$  is given  $n_2$ .

What knowledge contributes to the plausibility of the conclusions of the rules in Experiments 1 and 2? Said differently, what factors account for the *total variance in judgments of plausibility* ( $T$ ) among our subjects? We believe  $T$  has four additive components:

**subject variance**—the proportion of  $T$  due only to individual differences in subjects' knowledge, experience, motivation, and so on.

**item variance**—the proportion of  $T$  due only to differences in the concepts that instantiate  $n_1$ ,  $n_2$ ,  $n_3$  in the rule.

**between-rule variance**—the proportion of  $T$  due only to differences in the surface structures of rules.

**deep structure variance**—the proportion of  $T$  due only to whether deep structures are transitive, intransitive, or GPI structures.

Ideally, deep structure variance should account for the largest component of  $T$ . If 100% of  $T$  was due to deep structure variance, then transitivity and GPI would be perfect predictors of plausibility. In contrast, if a large fraction of  $T$  is due to item variance, then one needs to know the specific instantiation of a rule—the concepts in the test item—to predict its plausibility. Similarly, between-rule

variance represents the effect of knowing the surface structure of test items on one's ability to predict their plausibility. Subject variance represents the limit of our ability to predict plausibility.

For transitive and intransitive rules, and to a lesser extent for GPI rules, deep structure variance accounts for a large fraction of  $T$ . For all test items with these structural characteristics, our predictions of plausibility will be correct for 77% of transitive items and 68% of GPI items; and our prediction of *implausibility* will be correct for 62% of intransitive items. Since these numbers are not 100%, the remaining variance in  $T$  must be due to the rule, item, and subject factors.

Preliminary estimates of between-rule variance (based on the  $\omega^2$  statistic [101, p. 485]) are 16% for transitive rules, 27% for intransitive rules, and 52% for GPI rules. That is, if a rule is transitive, then knowing *which* rule it is provides little additional information about the plausibility of items. However, this knowledge accounts for much of the variance in plausibility scores of intransitive and GPI items.

Estimates of item and subject variance show that item differences account for most of the remaining variance. Knowing a rule's instantiation improves our prediction of its plausibility far more than knowing which subject is making the judgement. Details of these analyses are given in [99].

#### 14.3.4.4 Summary

Our results suggest that plausible inference rules can automatically be derived from the relations in knowledge bases and predict judgments of plausibility for the conclusions of these rules. Two structural factors (transitivity and GPI) correctly predict plausibility 77% and 68% of the time. No knowledge is required to apply these criteria. Greater accuracy requires more knowledge, particularly knowledge about the specific rules and the concepts that instantiate them; but because we could not accurately estimate the contribution of individual differences among our subjects to  $T$ , we do not know the limit on the accuracy of our predictions.

Our experiments relied on the GRANT KB, which was built for a different purpose. Although our results are limited to this knowledge base, we believe they are more general, because the surface relations in the GRANT KB are common, and because  $h$  and  $t$  are general semantic components, and because transitivity and GPI are common structural characteristics. But further work is required to *prove* the generality of our results.

Our goal was to develop methods to support graceful degradation and knowledge engineering. Clearly, these purposes are not met if plausible inference rules require masses of knowledge to judge their conclusions. We are very encouraged by

the relatively high accuracy of criteria that require no knowledge, and by the fact that our accuracy is higher for plausible rules than for implausible ones.

#### 14.3.5 Future Directions

The future research in this project will involve three parallel efforts:

1. Continuing the development of a formal framework for plausible inference. A goal for this framework will be that previous approaches to plausibility for retrieval are subsumed by the new approach. For example, previous probabilistic models of IR have used networks to model dependencies between terms used to represent document content. Bayesian inference nets can represent these dependencies and, potentially, other sources of evidence. The problem is to show that a single framework can be used to combine evidence from statistical (word-based) techniques, NLP, domain knowledge bases, and other sources such as citations. The work started on inference networks appears very promising, but many computational details remain to be worked out before a practical system is produced.
2. Completing the implementation of NLP techniques in the ADRENAL system. This provides the ability to use evidence from NLP for text objects. We are currently finishing the implementation of the parser and have built a knowledge base of frames representing technical concepts. The next stage is to implement the semantic transformation component that will produce the frame-based representations of queries and documents. We have also been studying the use of machine-readable lexicons as a source of evidence for retrieval. This NLP work constitutes an important effort in its own right as well as contributing to the plausible inference project. The ADRENAL system is potentially the first practical application of NLP techniques to IR.
3. Build a testbed based on GRANT knowledge base. To do this, we must process the original texts used to construct the GRANT knowledge base of research proposals. The processing will be done both with simple statistical techniques and with the NLP techniques described in step 2. We will also obtain the manually indexed version of the collection as an alternative source of evidence for retrieval. In order to do retrieval experiments, we must obtain a new set of queries with associated relevance judgments. We may also expand the CACM text collection.

Once these three efforts have been completed, a comprehensive series of experiments can be carried out to answer the major questions raised in this report. These include:

- Experiments to show that the plausible inference framework developed produces effective results. These experiments involve comparing ad hoc combinations of simple forms of evidence with combinations specified by the formal framework. For example, the experiments reported in [5], used a simple method of combining evidence based on a spreading activation model. Significant differences in performance resulted from variations in the combining functions used. The combining functions specified by the formal framework should be compared with these results to establish that it is appropriate for retrieval. The simple forms of evidence used in these experiments will also be a useful test of whether the form of estimation of probabilities or plausibility used in the formal framework are practical.
- Experiments to investigate the effectiveness of multiple sources of evidence. Once we have established an appropriate computational framework for plausible inference, it remains to be shown that multiple sources of evidence can reliably be used to improve performance beyond that possible with any single source, particularly sophisticated statistical retrieval strategies. The GRANT knowledge base allows us to study two very important sources of evidence; NLP and domain knowledge. In both cases, previous experiments have not established the utility of these forms of evidence. In order to demonstrate that they are useful, we must show that not only there is a performance improvement, but also that the improvement is large enough to outweigh the cost of NLP and generating domain knowledge. Results from these experiments will be crucial in influencing the future development of systems used for retrieval of complex objects.
- Experiments to show the effect of derived plausible inferences on retrieval. This is a continuation of the work reported by Cohen and Loisel [99] with an emphasis on incorporating the new inference rules in the retrieval process.

## References

- [1] Belkin, N. and Croft, W.B., 1987. "Retrieval Techniques". *Annual Review of Information Science and Technology*, Edited by M.E. Williams, Elsevier Science Publishers, 22, 110-145, 1987.
- [2] Cohen, P. , Davis, A. , Day, D. , Greenberg, M. , Kjeldsen, R. , Lander, S. , and Loiselle, C. 1985. *Representativeness and Uncertainty in Classification Systems* *AI Magazine*, 6(3), 136-149.
- [3] Cohen, P. and Kjeldsen, R. 1987. "Information Retrieval by Constrained Spreading Activation in Semantic Networks". *Information Processing and Management*, 23(4).
- [4] Croft, W. B.; Thompson, R., 1987. "I<sup>3</sup>R: A New Approach to the Design of Document Retrieval Systems". *Journal of the American Society for Information Science*, 38, 389-404, 1987.
- [5] Croft, W.B.; Lucia, T.J.; Cringean, J.; Willett, P., 1989. "Retrieving Documents by Plausible Inference: An Experimental Study". *Information Processing and Management*, (to appear).
- [6] Gallaire, H.; Minker, J.; Nicols, J., 1984. "Logic and Databases: A Deductive Approach", *ACM Computer Surveys*, 16, 153-186.
- [7] Katzer, J.; McGill, M.; Tessier, J.A.; Frakes, W.; Dasgupta, P., 1982. "A Study of the Overlap among Document Representations", *Information Technology*, 1, 261-274.
- [8] Lenat, Doug; Prakash, Mayank; and Shepherd, Mary. CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks. *AI Magazine*, 6(4), 65-85.
- [9] Lewis, D.; Croft, W.B.; Bhandaru, N., 1989. "Language Oriented Information Retrieval", *International Journal of Intelligent Systems*, (to appear).
- [10] Nilsson, Nils J. , 1984. Probabilistic Logic. *SRI AI Center Technical Note 321*, SRI International, Menlo Park, CA.
- [11] Pearl, J., 1989. *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann.

- [12] Van Rijsbergen, C. J., 1979. *Information Retrieval*. Second Edition. Butterworths, London.
- [13] Van Rijsbergen, C.J., 1986. "A Non-Classical Logic for Information Retrieval". *Computer Journal*, 29, 481-485
- [14] Salton, G. and McGill, M., 1983. *An Introduction to Modern Information Retrieval*, McGraw-Hill, New York.
- [15] Spiegelhalter, D.J., 1986. "A statistical view of uncertainty in expert systems". *Artificial Intelligence and Statistics*, 17-55, Addison Wesley.
- [16] Turner, Raymond, 1984. *Logics for Artificial Intelligence*. Chichester: Ellis Howard Limited.
- [17] Weyer, S.A. and Borning A.H., 1985. "A Prototype Electronic Encyclopedia". *ACM Transactions on Office Information Systems*, 3, 2-21.
- [18] Zadeh, L.A., 1975. Fuzzy logic and approximate reasoning. *Synthese*, 30, 407-428.
- [19] Croft, W.B. and Harper, D.J. 1979. "Using Probabilistic Models of Document Retrieval Without Relevance Information", *Journal of Documentation*, 35, 285-295 (1979).
- [20] Croft, W.B.; Lewis, D.. 1987. "An Approach to Natural Language Processing for Document Retrieval", *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 26-32, New Orleans, 1987.
- [21] Croft, W.B. and Thompson, R., 1984. "The Use of Adaptive Mechanisms for Selection of Search Strategies in Document Retrieval Systems", *Proceedings of the ACM/BCS International Conference on Research and Development in Information Retrieval*, 95-110.
- [22] El-Hamdouchi, A. and Willett, P., 1987. 'Techniques for the Measurement of Clustering Tendency in Document Retrieval Systems', *Journal of Information Science*, 13, 361-365.
- [23] Fox, E.A, Nunn, G.L., and Lee, W.C., 1988. 'Coefficients for Combining Concept Classes in a Collection', *Proceedings of the 10th SIGIR Conference on Research and Development in Information Retrieval*, Grenoble, France, 291-307.



- [24] Griffiths, A., Luckhurst, H.C. and Willett, P., 1986. 'Using Interdocument Similarity in Document Retrieval Systems', *Journal of the American Society for Information Science*, 37, 3-11.
- [25] McCall, F.M. and Willett, P., 1986. "Criteria for the Selection of Search Strategies in Best Match Document Retrieval Systems". *International Journal of Man-Machine Studies*, 25, 317-326.
- [26] Thompson, R. and Croft, W.B., 1989. "Support for Browsing in an Intelligent Text Retrieval System", *International Journal of Man-Machine Studies*, (to appear).
- [27] Tong, R.M. et al., 1987. "Conceptual Information Retrieval using RUBRIC". *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 247-253.
- [28] Willett, P., 1988. 'Recent Trends in Hierarchic Document Clustering: A Critical Review', *Information Processing and Management*, 24 (5), 577-598.
- [29] Croft, W.B.; Krovetz, R. "Interactive Retrieval of Office Documents", *Proceedings of the ACM Conference on Office Information Systems*, 228-235, (1988).
- [30] Gardarin, G.; Valduriez, P. *Relational Databases and Knowledge Bases*. Addison Wesley, Reading, MA., 1989.
- [31] Motro, A. "VAGUE: A User Interface to Relational Databases that Permits Vague Queries". *ACM Transactions on Office Information Systems*, 6, 187-214, 1988.
- [32] Robertson, S.E. "The Probability Ranking Principle in IR", *Journal of Documentation*, 33, 294-304, 1977.
- [33] Woelk, D.; Kim, W.; Luther, W. "An Object-Oriented Approach to Multimedia Databases". *Proceedings of SIGMOD-86*, 311-325, 1986.
- [34] Gerald Salton. Another look at automatic text-retrieval systems. *Communications of the ACM*, 29(7):648-656, July 1986.
- [35] Martin Dillon and Ann S. Gray. FASIT: A fully automatic syntactically based indexing system. *Journal of the American Society for Information Science*, 34(2):99-108, March 1983.

- [36] Donald J. Hillman and Andrew J. Kasarda. The LEADER retrieval system. In *Spring Joint Computer Conference*, pages 447-455. AFIPS, 1969.
- [37] N. Fuhr and G. E. Knorz. Retrieval test evaluation of a rule based automatic indexing (AIR/PHYS). In C. J. van Rijsbergen, editor, *Research and Development in Information Retrieval: Proceedings of the Third Joint BCS and ACM Symposium*, pages 391-408, Cambridge, July 2-6 1984. Cambridge University Press.
- [38] Paul M. Mott, David L. Waltz, Howard L. Resnikoff, and George G. Robertson. Automatic indexing of text. Technical Report 86-1, Thinking Machines Corporation, January 1986.
- [39] Lois L. Earl. Experiments in automatic extracting and indexing. *Information Storage and Retrieval*, 6:313-334, 1970.
- [40] Dario DeJaco and Gianluca Garbolino. An information retrieval system based on artificial intelligence techniques. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 214-220, 1986.
- [41] Alan F. Smeaton. Incorporating syntactic information into a document retrieval strategy: an investigation. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 103-113, 1986.
- [42] W. Bruce Croft. Boolean queries and term dependencies in probabilistic retrieval models. *Journal of the American Society for Information Science*, 37(2):71-77, 1986.
- [43] Joel L. Fagan. *Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and Non-Syntactic Methods*. PhD thesis, Department of Computer Science, Cornell University, September 1987.
- [44] Gerald Salton. *Automatic Information Organization and Retrieval*. McGraw-Hill Book Company, New York, 1968.
- [45] Y. Chiaramella and B. Defude. A prototype of an intelligent system for information retrieval: IOTA. *Information Processing and Management*, 23(4):285-303, 1987. Special Issue on Artificial Intelligence and Information Retrieval.
- [46] Marie-France Bruandet. Outline of a knowledge base model for an intelligent information retrieval system. In C. T. Yu and C. J. van Rijsbergen, editors, *Tenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 33-43, June 1987.

- [47] Catherine Berrut and Patrick Palmer. Solving grammatical ambiguities within a surface syntactical parser for automatic indexing. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 123-130, 1986.
- [48] G. Thurmair. A common architecture for different text processing techniques in an information retrieval environment. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 138-143, 1986.
- [49] G. Salton and M. E. Lesk. Computer evaluation of indexing and text processing. *Journal of the Association for Computing Machinery*, 15(1):8-36, 1968.
- [50] Edward A. Fox. Lexical relations: Enhancing effectiveness of information retrieval systems. *SIGIR Forum*, XV(3):5-36, 1980.
- [51] Edward A. Fox, J. Terry Nutter, Thomas Ahlswede, Martha Evens, and Judith Markowitz. Building a large thesaurus for information retrieval. In *Second Conference on Applied Natural Language Processing*, pages 101-108, February 1988.
- [52] Ronald J. Brachman and Hector J. Levesque, editors. *Readings in Knowledge Representation*. Morgan Kaufmann Publishers, Inc., Los Altos, California, 1985.
- [53] K. Sparck Jones and J. I. Tait. Automatic search term variant generation. *Journal of Documentation*, 40(1):50-66, March 1984.
- [54] Udo Hahn and Ulrich Reimer. Topic essentials. Technical Report TOPIC-19/86, Universitat Konstanz, Konstanz, April 1986.
- [55] Ulrich Thiel and Rainer Hammwohner. Informational zooming: An interaction model for the graphical access to text knowledge bases. In C. T. Yu and C. J. van Rijsbergen, editors, *Tenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 45-56, June 1987.
- [56] Richard H. Granger, Chris J. Staros, Gregory B. Taylor, and Rika Yoshii. Scruffy text understanding: Design and implementation of the NOMAD system. In *Conference on Applied Natural Language Processing*, pages 104-106, February 1983.
- [57] James R. Cowie. Automatic analysis of descriptive texts. In *Conference on Applied Natural Language Processing*, pages 117-123, February 1983.

- [58] Naomi Sager. *Natural Language Information Processing: A Computer Grammar of English and Its Applications*. Addison-Wesley, Reading, Massachusetts, 1981.
- [59] Sheryl R. Young and Philip J. Hayes. Automatic classification and summarization of banking telexes. In *The Second Conference on Artificial Intelligence Applications*, pages 402-408. IEEE Computer Society, December 1985.
- [60] Steven Lytinen and Anatole Gershman. ATRANS: automatic processing of money transfer messages. In *AAAI-86*, pages 1089-1093, 1986.
- [61] Larry H. Reeker, Elena M. Zamora, and Paul E. Blower. Specialized information extraction: Automatic chemical reaction coding from English descriptions. In *Conference on Applied Natural Language Processing*, pages 109-116, 1983.
- [62] Lisa F. Rau and Paul S. Jacobs. Integrating top-down and bottom-up strategies in a text processing system. In *Second Conference on Applied Natural Language Processing*, pages 129-135, February 1988.
- [63] David Allport. The TICC: Parsing interesting text. In *Second Conference on Applied Natural Language Processing*, pages 211-218, February 1988.
- [64] Fujio Nishida and Shinobu Takamatsu. Structured-information extraction from patent-claim sentences. *Information Processing and Management*, 18(1):1-13, 1982.
- [65] Gerald DeJong. An overview of the FRUMP system. In Wendy G. Lehnert and Martin H. Ringle, editors, *Strategies for Natural Language Processing*, pages 149-176. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1982.
- [66] Michael Mauldin, Jaime Carbonell, and Richard Thomason. Beyond the keyword barrier: Knowledge-based information retrieval. In *29th Annual Conference of the National Federation of Abstracting and Information Services*. Elsevier Press, 1987.
- [67] Richard M. Tong, Lee A. Appelbaum, Victor N. Askman, and James F. Cunningham. Conceptual information retrieval using RUBRIC. In C. T. Yu and C. J. van Rijsbergen, editors, *Tenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 247-253, June 1987.

- [68] Richard M. Tong and Lee A. Appelbaum. Conceptual information retrieval from full-text. In *RIAO 88 : User-Oriented Content-Based Text and Image Handling*, pages 899-909, 1988.
- [69] Philip J. Hayes, Laura E. Knecht, and Monica J. Cellio. A news story categorization system. In *Second Conference on Applied Natural Language Processing*, pages 9-17, February 1988.
- [70] C. W. Cleverdon and E. M. Keen. Aslib-Cranfield research project. Vol. 2, Test results. Technical report, Cranfield Institute of Technology, Cranfield, England, 1966.
- [71] Jean Aitchison and Cyril Cleverdon. A report on the test of the index of metallurgical literature of Western Reserve University. Technical report, The College of Aeronautics, Cranfield, Cranfield, England, October 1963.
- [72] J. C. Gardin. *SYNTOL*, volume II of *Rutgers Series on Systems for the Intellectual Organization of Information*. Graduate School of Library Service, Rutgers, 1965.
- [73] Derek Austin. PRECIS in a multilingual context. *Libri*, 26(1):1-37, 1976.
- [74] D. B. McCarn. Medline: An introduction to on-line searching. *Journal of the American Society for Information Science*, 31(3):181-192, May 1980.
- [75] J. Farradane. Relational indexing. part I. *Journal of Information Science*, 1:267-276, 1980.
- [76] J. Farradane. Relational indexing. part II. *Journal of Information Science*, 1:313-324, 1980.
- [77] J. Farradane, J. M. Russell, and P. A. Yates-Mercer. Problems in information retrieval: Logical jumps in the expression of information. *Information Storage and Retrieval*, 9:65-77, 1973.
- [78] Glenn D. Rennels, Edward H. Shortliffe, Frank E. Stockdale, and Perry L. Miller. Reasoning from the clinical literature: The Roundsman system. Memo KSL-86-5, Medical Computer Science Group, Stanford University School of Medicine, January 1986.
- [79] Gian Piero Zarri. Some remarks about the inference techniques of RESEDA, an "intelligent" information retrieval system. In C. J. van Rijsbergen, editor, *Research and Development in Information Retrieval: Proceedings of the*

*Third Joint BCS and ACM Symposium*, pages 281-300, Cambridge, July 1984. Cambridge University Press.

- [80] Robert F. Simmons. A text knowledge base from the AI handbook. *Information Processing and Management*, 23(4):321-339, 1987. Special Issue on Artificial Intelligence and Information Retrieval.
- [81] Deb Krawczak, Philip J. Smith, and Steven J. Shute. EP-X: a demonstration of semantically-based search of bibliographic databases. In *Tenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 263-271, 1987.
- [82] DeeAnn Emmel Lewis. *Case Grammar and Functional Relations in Aboutness Recognition and Relevance Decision-Making in The Bibliographic Retrieval Environment*. PhD thesis, School of Library and Information Science, Faculty of Graduate Studies, The University of Western Ontario, July 1984.
- [83] Jamie G. Carbonell, David A. Evans, Dana S. Scott, and Richard H. Thomason. On the design of biomedical knowledge bases. In R. Salamon, B. Blum, and M. Jorgenson, editors, *Medinfo86: Proceedings of the Fifth Conference on Medical Informatics*, pages 37-41, Amsterdam, 1986. Elsevier Science Publishers.
- [84] Paul S. Jacobs. Language analysis in not-so-limited domains. In *Fall Joint Computer Conference*, Dallas, TX, 1986.
- [85] Paul S. Jacobs. A knowledge framework for natural language analysis. In *Tenth International Joint Conference on Artificial Intelligence*, pages 675-678, August 23-28 1987.
- [86] James Allen. *Natural Language Understanding*. The Benjamin/Cummings Publishing Company, Inc., Menlo Park, California, 1987.
- [87] Robert Wilensky. Some problems and proposals for knowledge representation. Technical Report UCB/CSD87/351, EECS, University of California, Berkeley, May 1987.
- [88] David J. Besemer and Paul S. Jacobs. FLUSH: a flexible lexicon design. In *25th Annual Meeting of the Association for Computational Linguistics*, pages 186-192, 1987.
- [89] K. Sparck Jones and C. J. van Rijsbergen. Information retrieval test collections. *Journal of Documentation*, 32(1):59-75, 1976.

- [90] Samuel Bayer, Leonard Joseph, and Candace Kalish. Grammatical relations as the basis for natural language parsing and text understanding. In *Ninth International Joint Conference on Artificial Intelligence*, pages 788-790, August 18-23 1985.
- [91] Henry Thompson. MCHART: a flexible, modular chart parsing system. In *AAAI-83*, pages 408-410, 1983.
- [92] Eugene Charniak, Christopher K. Riesbeck, Drew V. McDermott, and James R. Meehan. *Artificial Intelligence Programming*. Lawrence Erlbaum Associates, Hillsdale, NJ, second edition, 1987.
- [93] Martin Kay. Algorithm schemata and data structures in syntactic processing. Technical Report CSL-80-12, Xerox Palo Alto Research Center, Palo Alto, California 94304, October 1980.
- [94] Kent Wittenburg. Parsing as heuristic graph search. Technical Report HI-075-86, Microelectronics and Computer Technology Corporation. Human Interface Program, 6 March 1986.
- [95] Bran Boguraev and Ted Briscoe. Large lexicons for natural language processing: Utilising the grammar coding system of LDOCE. *Computational Linguistics*, 13(3-4):203-218, 1987. Special Issue on the Lexicon.
- [96] Steven J. DeRose. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 14(1):31-39, 1988.
- [97] Kenneth Ward Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Second Conference on Applied Natural Language Processing*, pages 136-143, February 1988.
- [98] Ronald J. Brachman. "I lied about the trees" or, defaults and definitions in knowledge representation. *AI Magazine*, 6(3):80-93, Fall 1985.
- [99] Paul R. Cohen and Cynthia L. Loiselle. *Explorations in the Structure of Plausible Inference Rules for Large Knowledge Bases*. Technical Report 88-54, University of Massachusetts, Amherst, MA, 1988.
- [100] A. Collins, E. Warnock, N. Aiello, and M. Miller. Reasoning from incomplete knowledge. In D. G. Bobrow and A. Collins, editors, *Representation and Understanding*, Academic Press, New York, 1975.

- [101] Willim L. Hays. *Statistics for the Social Sciences*. Holt, Rinehart, and Winston, second edition, 1973.
- [102] D. B. Lenat and E. Feigenbaum. On the thresholds of knowledge. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 1173–1182, Milan, Italy, 1987.





# *MISSION of Rome Air Development Center*

*RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control, Communications and Intelligence (C<sup>3</sup>I) activities. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C<sup>3</sup>I systems. The areas of technical competence include communications, command and control, battle management information processing, surveillance sensors, intelligence data collection and handling, solid state sciences, electromagnetics, and propagation, and electronic reliability/maintainability and compatibility.*